Laboratoire de régulation et d'automatique

Projet de semestre 2003

Contrôle d'un système de tracking optique

Ecole d'ingénieur de l'Arc Jurassien
Dpt. Electricité

Professeur: **Moncef Justin Lahlou** (lahlou@eicn.ch)
Etudiant: **Marguet Julien** (marguet@eicn.ch)

1. Résumé

Ce projet a pour objectif principal de mettre au point un système dit de tracking optique. Une partie de ce projet a déjà été réalisée au laboratoire d'optique. Ainsi deux axes de rotation entraînés par 2 moteurs, permettant de régler l'azimut et l'élévation d'un capteur optique, ont été mis à disposition pour ce projet.
Le but de ce système est d'aligner une antenne directionnelle qui serait fixée au capteur optique, vers une source infrarouge qui serait placée à plusieurs dizaines de mètre (voir plus). Ce système serait couplé à une caméra, les signaux de la caméra serait alors envoyé via cette antenne.
Il restait pour ma part à développer une carte afin de traiter les signaux provenant du capteur optique, de les transmettre à un microcontrôleur afin de réguler la position du moteur azimut pour centrer le capteur optique en face de la source infrarouge et finalement de réaliser un circuit pour la partie puissance de ce moteur.
Le microcontrôleur utilisé est un 80C167CR de chez ST Microelectronics.
Le projet s'est relativement bien déroulé. Il n'était cependant pas prévu au début du projet qu'il faille traiter les signaux provenant du capteur optique. Et c'est cette partie qui s'est avérée la plus longue et la plus importante à mettre en œuvre.
Le projet, au point actuel, permet d'aligner correctement le capteur optique en face de la source infrarouge selon l'azimut et ce à plus de 10 mètres, cette distance dépendant très fortement du filtrage qu'on applique.
Le moteur permettant d'aligner selon l'élévation était en panne et fut réparer 3 semaines avant la fin de ce projet, cela ne m'a pas laissé le temps de mettre en place la partie électronique et le programme pour réguler la position selon l'élévation. De plus il semble qu'une des photodiodes du capteur optique soit hors d'usage.

2. Abstract

The goal of this project was to develop a system known as optical tracking. A part of this project was already developped by the laboratory of optics. Thus two axes of rotation driven by 2 motors , making it possible to regulate the azimuth and the rise of an optical sensor, were placed at the disposal for this project.
The goal of this system is to align an antenna which would be fixed with the optical sensor, towards an infra-red source which would be placed at several tens of meter (more if possible). My goal was to develop and to analyse the signals coming from the optical sensor, to transmit them to a microcontrolor in order to regulate a motor and finally to developp a circuit to give some pwer to this motor. The microcontrolor was a 80C167CR from Microelectronics.
The project proceeded relatively well. However it wasn't envisaged at the beginning of the project that the signals coming from the optical sensor have to be analysed. And it is this part that had been the longest and the most significant to do.
The project, at the moment, makes it possible to align correctly the azimuth of the optical sensor in direction of the infra-red source and this with more than 10 meters, this distance depending strongly of the filtering which is applied.
The motor making it possible to align according to the rise was broken and it was repaired 3 weeks before the end of this project, that gives me not enough time to end the electronic part and the program to control the position according to rise. Moreover it seems that one of the photodiodes of the optical sensor is out of use.

Table des matières

1. Résumé 2
2. Abstract 2
3. Cahier des charges 4
3.1. Objectifs 4
3.2. Ressources 4
3.3. Références 5
3.4. Planning et déroulement du projet 6
4. Introduction 7
5. Le microcontrôleur C167CR 8
5.1. La carte de développement Phytec 8
5.1.1. Les cavaliers 9
5.1.2. Le kit de développement 9
6. Utilisation de Tasking EDE 10
6.1. Création d'un projet 10
7. L'électronique 18
7.1. Etude sur la source infrarouge 18
7.2. Etude sur le capteur optique 20
7.2.1. Les photodiodes 22
7.2.2. Montage de base pour amplifier les signaux issus de photodiode 23
7.3. Traitement du signal issu d'une photodiode 25
7.3.1. Stratégie de traitement du signal 25
7.3.2. La pré amplification 28
7.3.2.1. Simulation Pspice 30
7.3.3. Le filtrage 30
7.3.4. Le redressement 30
7.3.5. Extraction de la composante continue 30
7.3.6. Limitation du signal 31
7.3.7. Pilotage du moteur azimut 31
7.3.7.1. Le driver 34
7.3.7.2. Le microcontrôleur en mode nomade 34
7.3.7.3. Le montage 34
7.3.8. Circuit final 35
7.3.9. PCAD 36
7.3.9.1. Les librairies 36
7.3.9.2. Le schéma 40
7.3.9.3. Le routage 47
7.3.10. Amélioration 48
7.3.11. Problèmes rencontrés 56
8. Le programme 57
8.1. Acquisition d'un signal 57
8.2. Utilisation d'une unité PWM 59
8.3. Le programme final 63
8.4. Problèmes rencontrés 69
9. Conclusion 69
10. Annexes 70

3. Cahier des charges

3.1. Objectifs

- Etude du projet réalisé par le laboratoire d'optique et en comprendre le fonctionnement afin d'établir une stratégie pour la régulation et le placement du capteur optique, étude du capteur optique et de ses photodiodes. Etude de la modulation générée par la source infrarouge.
- Installation du système de développement du microcontrôleur C167 de chez Microelectronics, comprenant les logiciels de développement Tasking (compilateur C, assembleur) et la carte de développement Phytec équipée du microcontrôleur C167CR, en prendre connaissance et expérimentation de divers programmes.
- Etude, simulation (sous PSPICE) et réalisation d'une carte électronique pour traiter les signaux provenant du capteur optique afin de les rendre compatible avec les entrées analogiques disponibles sur le microcontrôleur.
- Etablissement d'une stratégie pour contrôler les moteurs selon les possibilités offertes par le microcontrôleur.
- Acquisition des signaux du capteur optique par le microcontrôleur, établissement d'une stratégie de régulation du moteur selon l'azimut, implémentation et test pratique du programme, génération de signaux PWM
- Pilotage d'un driver de moteur par le microcontrôleur.
- Optimisation de la carte de traitement des signaux provenant du capteur optique, mise en place d'un filtre pour n'acquérir que les signaux provenant de la source infrarouge (fréquence de 40kHz).
- Utilisation du microcontrôleur en mode autonome (nomade).
- Développement sur PCAD des librairies et du routage nécessaire à la réalisation d'un circuit imprimé comportant le minimodule C1567CR, la partie traitant les signaux des photodiodes et le driver pour piloter le moteur.
- Ajout de la fonction pour le positionnement selon l'élévation si le temps le permet

3.2. Ressources

- **Humaine :**

Un professeur pour l'encadrement du projet (M. Lahlou).

- **Matérielles :**

Mise à disposition d'un PC + logiciels de développement Tasking + carte de développement Phytec + prototype comprenant l'axe de rotation, le capteur optique et la source infrarouge venant du laboratoire d'optique + éléments et moyens de développement électronique du laboratoire de régulation.

3.3. Références

- **Livres :**
  [1] Tasking Manuels de programmation du C167.
  [2] Phytec Documentation sur la carte de développement autour du C167.
- **Cours :**
  [3] M. Brandt Cours du moteur à courant continu, cours sur la PWM. (2001/2002)
  [4] M. Sandoz Cours de télécommunications sur les techniques de filtrage.
  [5] M. Amez-Droz Cours sur les AOP (2001/2002)

- **PDF :**
  [6] L6203 Caractéristiques techniques et notes d'application du driver L6203
  [7] TL084 Caractéristiques techniques et notes d'application de l'AOP TL084
  [8] TL082 Caractéristiques techniques et notes d'application de l'AOP TL082
  [9] LM6142 Caractéristiques techniques et notes d'application de l'AOP LM6142
  [10] MAX412 Caractéristiques techniques et notes d'application de l'AOP MAX412
  [11]Maxon A-max22 Caractéristiques techniques et notes d'application du moteur azimut
  [12]Minimodule C167 Caractéristiques techniques et notes d'application sur le minimodule C167
  [13]Famille C167 Caractéristiques techniques et notes d'application sur la famille de microcontrôleur C167

- **Rapports :**
  [14] Projet de diplôme 2002/2003 --- Carte de contrôle à µC pour minimoteur pas à pas, par Nathanaël Sunier
  [15] Etude sur les photodiodes et la modulation FSK Juin 2003 chez M. Sandoz

- **Internet**

3.4. Planning et déroulement du projet

Le projet se déroule de la mi-mars à la fin juin: les jeudis y sont exclusivement consacrés. Beaucoup de travail hors de cette journée y est aussi consacré.

- Le mois de mars a été consacré à l'étude du projet réalisé par le laboratoire d'optique.
- 2 semaines ont pleinement été consacrées à l'étude du langage C appliqué au domaine du microcontrôleur. (La programmation en langage C des microcontrôleurs n'a été qu'entrevue en cours à ce moment).
- Une fois la carte de développement Phytec reçue j'ai pu mettre en application certains programmes et ce pendant 2 semaines.
- Réalisation / simulation (Pspice) de la carte de traitement des signaux: cette phase s'est faite en 2 temps. Mi-avril à fin mai.
- Conception du programme de gestion de l'azimut: cette partie n'a pas pris énormément de temps: 3 semaines réparties sur toute la période du projet.
- Quelques jours nécessaires au test du driver du moteur azimut.
- PCAD: réalisation du circuit imprimé. 3 semaines

4. Introduction

Ce projet a pour objectif principal de mettre au point un prototype de contrôle permettant la régulation du positionnement d'un capteur optique en face d'une source infrarouge modulée (ON/OFF) à 40kHz.
Pour cela un moteur permet de régler l'azimut dans les 2 sens (horaire et anti-horaire).

La 1ére étape de ce projet sera de prendre en main le matériel mis à disposition: apprendre à utiliser le logiciel Tasking pour la programmation du µC (microcontrôleur), réaliser un programme simple avec ce µC.
Les principales fonctions du C167CR seront décrites, la disposition des cavaliers sur la carte de développement sera donnée.

Une fois cette étape effectuée ce rapport traitera de la partie électronique pour le traitement des signaux provenant du capteur optique et du circuit pilotant le moteur azimut.
Une sous partie a été consacrée à la simulation sous Pspice (logiciel que je ne connaissais pas) de ce circuit.
La création du circuit imprimé avec PCAD (logiciel que je ne connaissais pas) y est décrite.
Les améliorations qui pourraient être portées au circuit sont données ainsi que les principaux problèmes rencontrés.

Une partie portera sur le programme permettant d'acquérir les signaux des photodiodes du capteur optique. La régulation pour le contrôle du moteur en azimut y est expliquée. Tout comme la partie précédente des propositions d'amélioration y sont indiquées tout comme les principaux problèmes rencontrés.

5. Le microcontrôleur C167CR

Cette partie 5 traite de ce µC qui a été utilisé lors du projet. D'abord la première question qu'on pourrait se poser est pourquoi le C167CR ?
Ce µC a été utilisé lors du projet de diplôme 2002/2003 par M. Nathanaël Sunier et cette personne a pu me donner quelques conseils avant de quitter l'école. La raison de ce choix de µC remonte donc à son travail de diplôme. Ce µC a été choisi à son époque pour le fait qu'il dispose du bus CAN, qu'il est de conception récente, qu'il est disponible à la vente, qu'il offre une mémoire importante et qu'il dispose d'unités PWM utiles pour piloter un moteur.
Voici les principales caractéristiques de ce µC et du minimodule (Figure 1) de chez Phytec qui comporte ce µC:

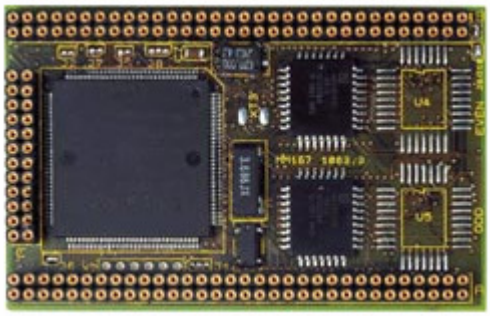


Figure 1: Minimodule Phytec

On peut voir le µC sur la figure précédente (partie gauche).

Caractéristique :
Processeur 16 bits (SAB C167-CR) avec chip CAN version 2.0B
Minimodule au format carte de crédit
Mémoire :
- 256kB SRAM
- 256kB Flash

Horloge à 20MHz.
Tous les signaux et ports du contrôleur sont étendus à des broches de largeur standard sur trois bords de la carte. Ces broches disposent d´un port d´entrées/sorties série bidirectionnel 16 bits, de 64 entrées/sorties et d´un convertisseur A/D 16 canaux à résolution 10 bits.
2 liaisons série RS-232 (dont 1 configurable en RS485)
Alimentation +5v /250mA

Ce minimodule est souvent utilisé dans les domaines utilisant beaucoup de mémoire, dans les applications utilisant le bus CAN.

5.1. La carte de développement Phytec

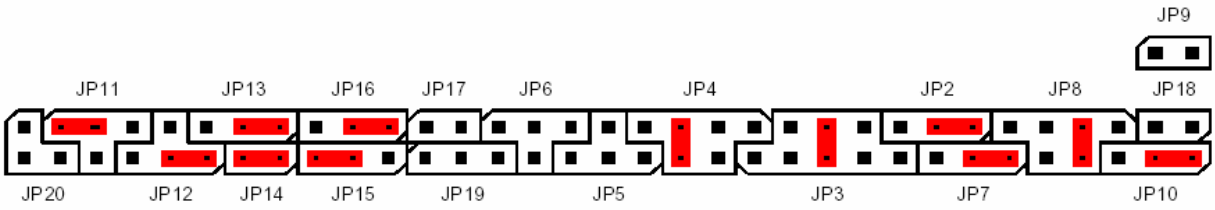
Cette carte a été utilisée pour le projet, elle reçoit le C167CR sur un connecteur. Cette carte de développement permet via un logiciel, de flasher la mémoire du C167CR. De plus via un logiciel (Crossview Pro) on peut connaître en temps réel les valeurs des différents registres…et ce en mode pas à pas (pour le débugage) ce qui est un grand avantage.
(Rq: il ne s'agit pas en réalité d'un vrai mode temps réel).

Figure 2: Carte de développement Phytec (Sans le minimodule C167CR)

Pour utiliser cette carte il a fallu installer le logiciel de développement Tasking. Ce logiciel reconnaît cette carte Phytec et son microcontrôleur.
Si le développement d'un programme venait à être fait avec un autre compilateur que Tasking, il faudrait compiler le programme avec certaines options (qui seront vue plus loin) afin de générer un fichier .hex compatible avec le C167CR. Un logiciel (FlashTools) permet de flasher la mémoire du C167CR (il dispose d'une interface graphique ou DOS de très simple utilisation).
Rq: lors de l'installation du logiciel il a fallu demander une clé au fabricant du logiciel. Celle-ci dépend de l'adresse de l'ordinateur sur le réseau.

5.1.1. Les cavaliers

Voici la disposition des cavaliers sur la carte de développement :



5.1.2. Le kit de développement

Celui-ci se compose de :

- module à microcontrôleur miniMODUL-167CR
- carte de développement pour une mise en œuvre rapide
- adaptateur secteur
- câble d'interface série pour raccordement au PC
- CD-Spektrum PHYTEC
- compilateur, assembleur et moniteur en version d´évaluation
- FlashTools
- manuel quickStart imprimé
- garantie de mise en œuvre

Le prix de ce kit est d'environ 230 €
Il est à noter que ce kit n'a été reçu qu'un mois après le début du projet. Mais le logiciel Tasking était déjà au laboratoire en version achetée, il suffisait de l'installer.

6. Utilisation de Tasking EDE

Il est à noter que Tasking et Phytec fournissent de nombreux exemples très utiles pour la compréhension du fonctionnement du C167CR notamment le fonctionnement du port série, des entrées sorties numériques analogiques, des convertisseurs (analogiques → numérique), du fonctionnement de la PWM…
La société Tasking fournit plusieurs logiciels dont une suite nommée Tasking EDE (EDE pour: Embedded Development Environment) groupant un compilateur C, C++ et assembleur. Cette suite possède une interface graphique. Une fois le logiciel installé j'ai procédé au test de la carte Phytec en suivant un exemple détaillé qu'il suffisait de suivre pas à pas.

6.1. Création d'un projet

La suite de cette partie décrit comment réaliser un projet permettant notamment d'allumer ou d'éteindre une led sur la carte Phytec aux fréquences voulues.

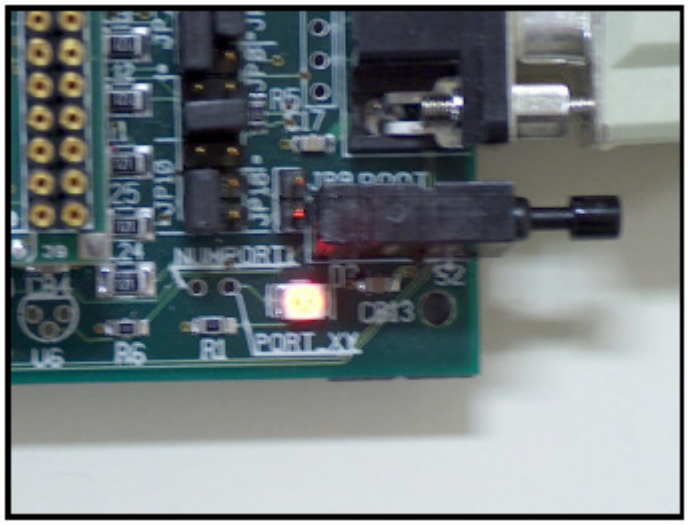
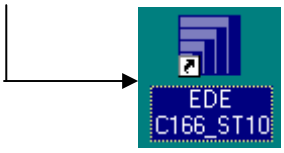
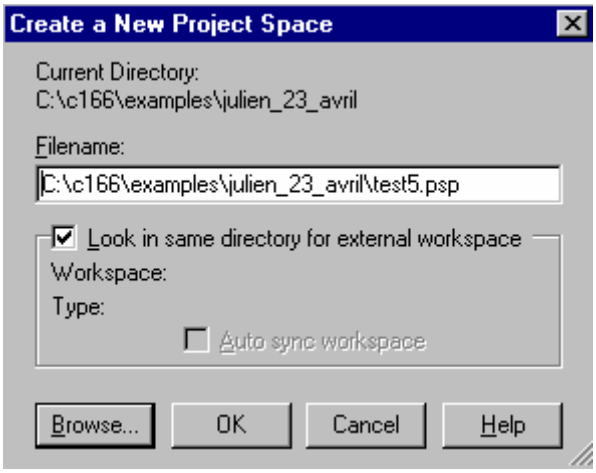


Figure 3: Led sur la carte Phytec

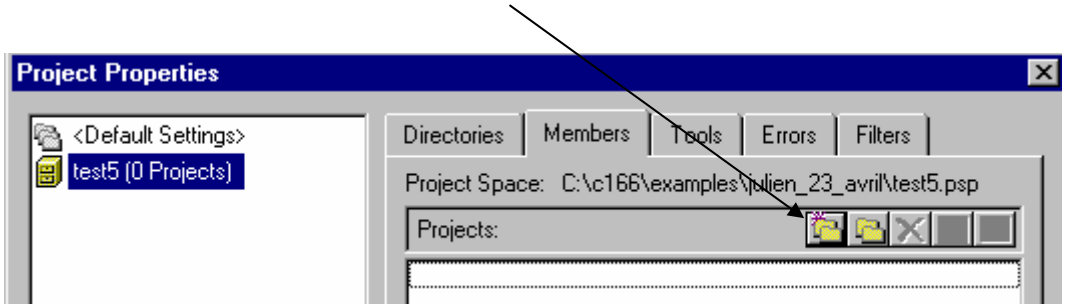
Etape 1: Ouverture de EDE (par le menu démarrer) ou directement sur l'icône.

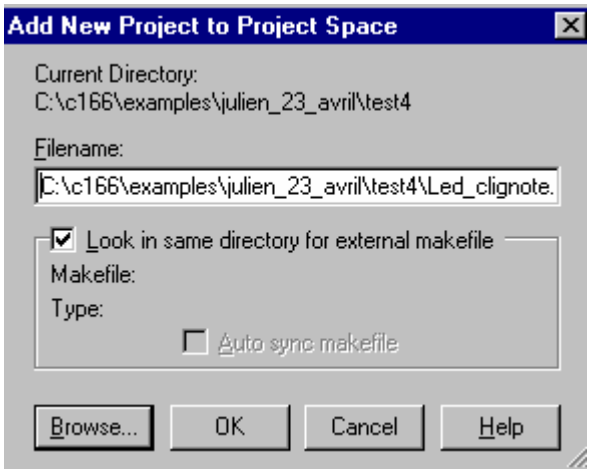


Pour créer le projet :
Aller dans : *Menu : Project Space -> New…*
Puis sélectionner le répertoire de destination et le nom de l'espace projet.



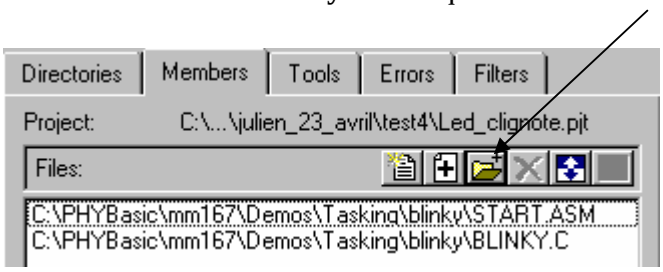
Valider, la fenêtre suivante apparaît :
L'étape suivante consiste à créer un projet dans cet espace de projet. Pour cela cliquer sur l'icône désigné par la flèche. Désigner un répertoire et nom de projet (ici: Led_clignote.pjt)





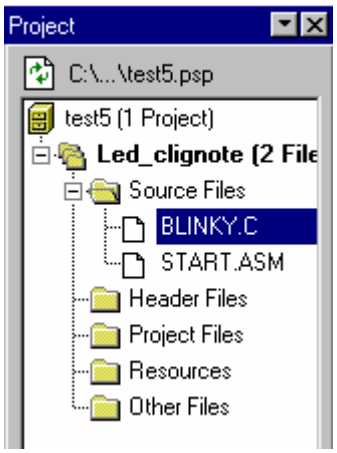
Valider par OK.

Il faut ajouter les fichiers sources : **Start.ASM** et **Blinky.c** en cliquant sur l'icône



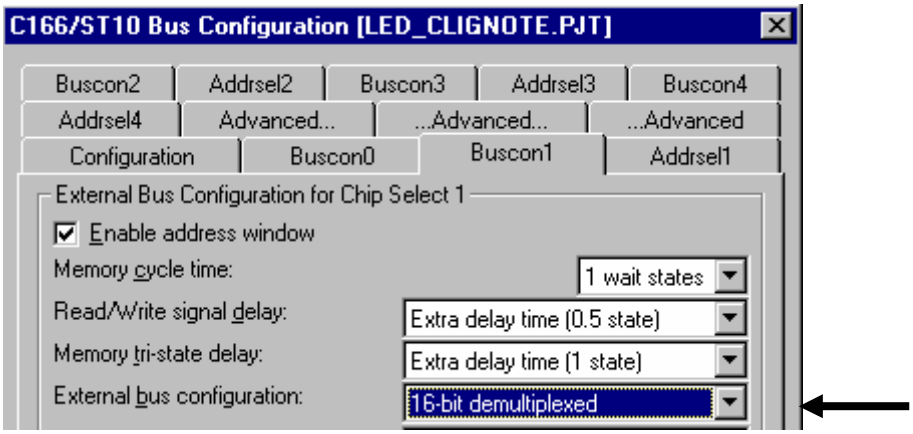
Start.asm doit être ajouté à chaque projet, il s'agit d'un fichier de Startup pour les programmes en langage C pour la famille de processeur C167.
Blinky.c contient le code source en langage C du progamme permettant de faire clignoter la led. Son contenu sera décrit plus loin.

Valider par OK.
On trouve ainsi cette disposition de fichier dans notre projet:

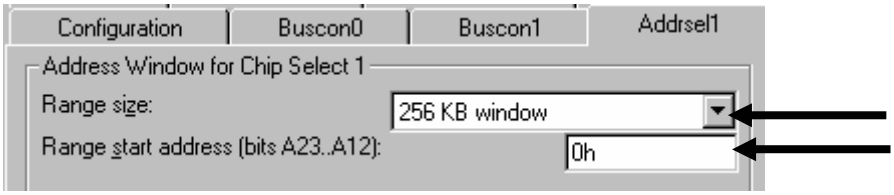


Dès lors il faut configurer EDE pour le faire fonctionner avec notre µC. Pour cela aller dans le menu : *EDE -> Project Option -> CPU -> C167CR.* Puis valider par OK

Ensuite aller dans le menu :
EDE -> Bus Configuration -> et faites les changements désignés par les flèches
Dans l'onglet Buscon1:

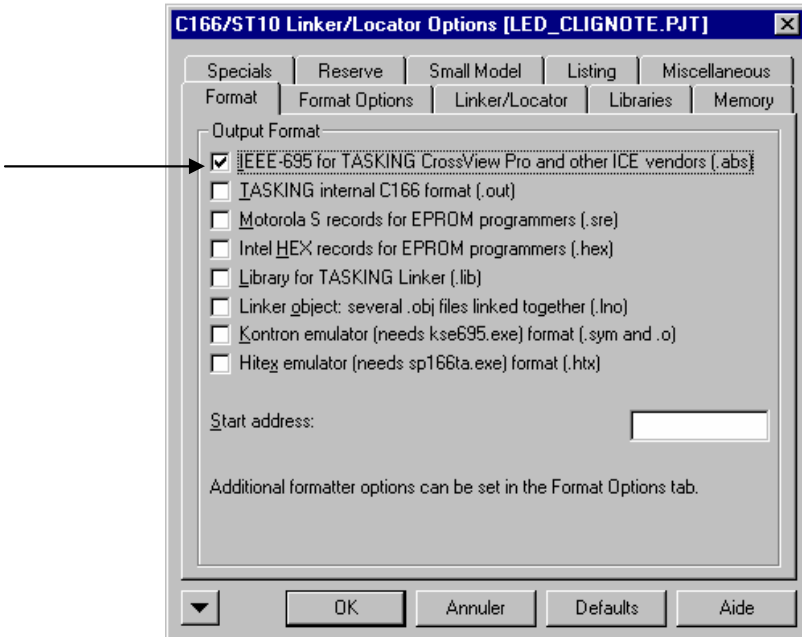


Dans l'onglet Addrsel1:



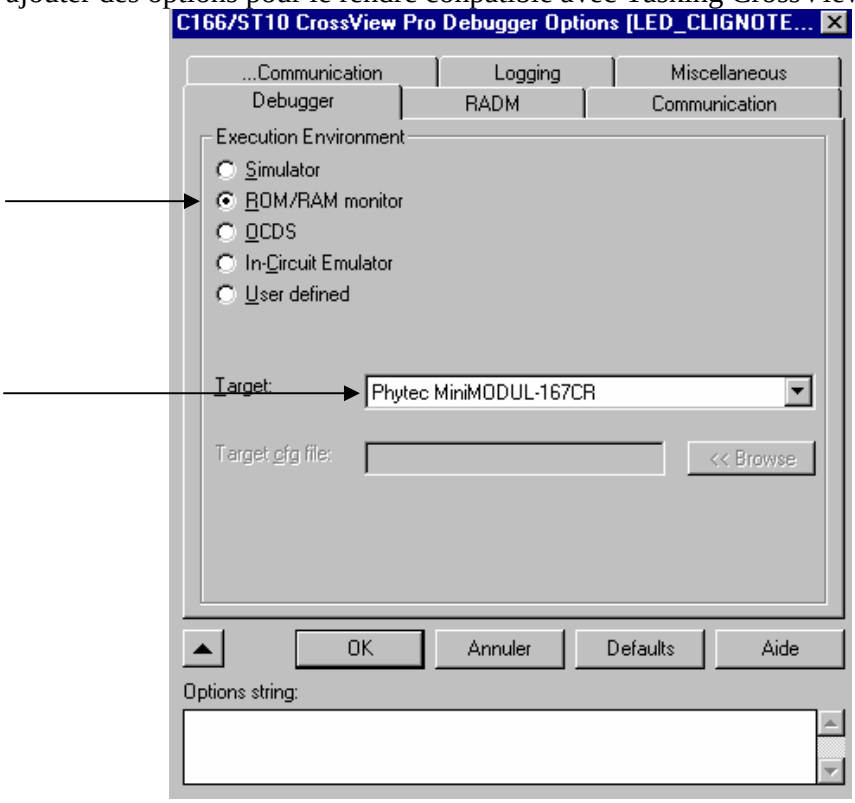
Valider par OK

Dans le menu EDE -> *Linker /Locator Options* : Faire les modifications suivantes :
Output Fomat :



Cela permet de choisir le format de sortie, on pourra ainsi utiliser le débuggeur Tasking CrossView Pro.
Small Model : Default

Il faut ensuite ajouter des options pour le rendre conpatible avec Tasking CrossView Pro.



Finalement on peut compiler et linker le projet : cliquer sur cette icône
Vérifier qu'il n'y a pas d'erreur en bas du logiciel. Si c'est le cas on peut flasher le programme dans le C167CR.
Pour ce, il faut mettre le minimodule Phytec en mode Bootstrap.
Appuyer sur les boutons :
Reset + Boot… et relâcher Reset en premier.

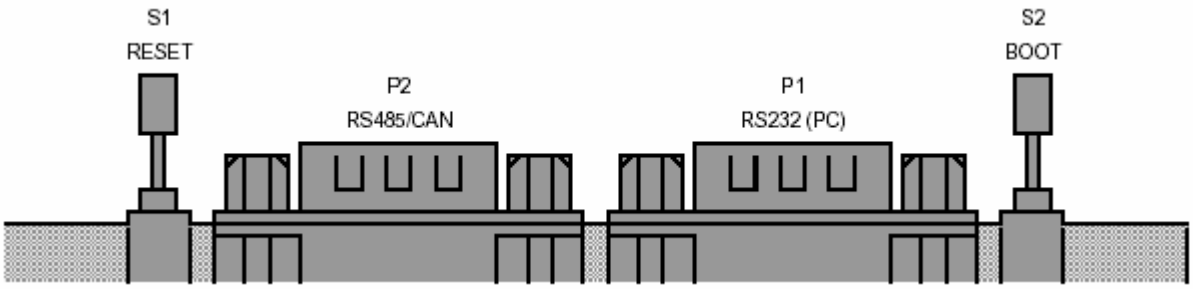
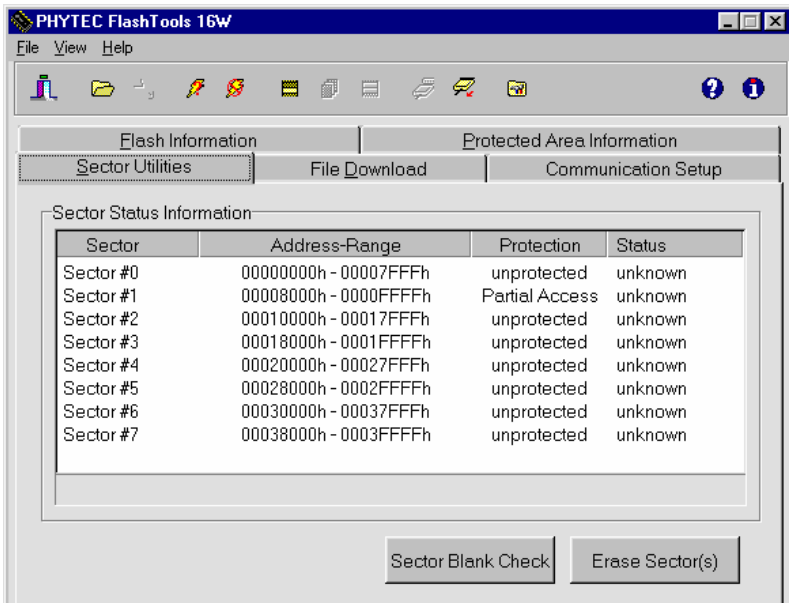


Figure 4 : Mode Bootstrap

On peut dès lors utiliser Crossview en cliquant sur son icône:
Le test en mode pas à pas est alors possible. On peut finalement constater le clignotement de la led si l'on se met en mode run.

Rq: il aurait été tout à fait possible de ne pas utiliser le débuggeur CrossView Pro. Pour cela il aurait fallu faire une autre configuration et choisir le format de sortie au format **Intel .HEX record.**

Le fichier .hex (disponible dans le répertoire courant du projet) aurait du être flashé dans le C167CR via le logiciel Flashtools. Encore une fois il ne faut pas oublier de se mettre en mode Bootstrap, puis ouvrir le software FlashTools16w.
Sélectionner le Minimodul 167 mode 1. Puis cliquer sur Connect : l'écran suivant apparaît :



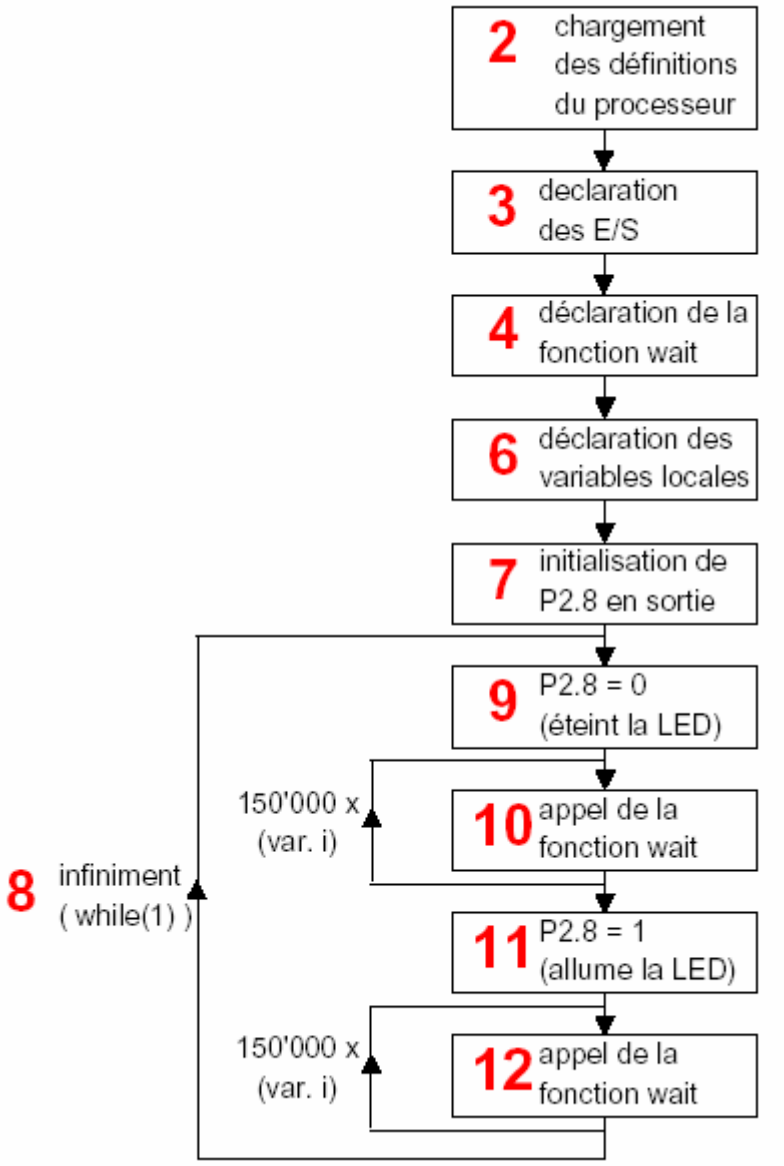
Il faut choisir tous les secteurs et les effacer. Dans l'onglet File Download télécharger le fichier .hex.. Il faut ensuite se déconnecter via l'icône disponible sur le logiciel. On peut faire clignoter la led. (Une fois le bouton S1 pressé la led clignote toujours).
Une légère étude sur le programme est présentée ci-dessous.

Fichier source de Blinky.c (copie issue du rapport de M. Sunier, code issu d'un exemple fournit par Phytec):

```
1  /* FLASH.C - LED Flasher for Phytecs Rapid Development Kits                */
2
3  #include <reg166.h>                    /* include 80C167 header file       */
4
5  _sfrbit P2_8       _atbit(P2,8);       /* declare portpin to use           */
6  _sfrbit DP2_8      _atbit(DP2,8);      /* declare direction bit to use     */
7
8  void wait (void)  {                    /* wait function                    */
9    return;                              /* only to delay for LED flashes    */
10 }
11
12 void main (void)  {
13   unsigned long i;                     /* delay var                        */
14
15   DP2_8 = 1;                           /* init direction to output         */
16
17   while (1) {                          /* loop forever                     */
18
19     P2_8 = 0;                          /* output to LED port               */
20     for (i=0; i<150000; i++) {         /* delay for 150000 counts          */
21       wait ();                         /* call wait function               */
22     }
23
24     P2_8 = 1;                          /* output to LED port               */
25     for (i=0; i<150000; i++) {         /* delay for 150000 counts          */
26       wait ();                         /* call wait function               */
27     }
28   }
29 }
30
31
```

Fonctionnement de ce programme:
La ligne 3 déclare les E/S (le port P2.8 : qui est en réalité la led).
La ligne 4 définit la direction de ce bit comme port de sortie
A la ligne 12 commence le programme principal. Celui-ci s'effectue à l'infini:
La boucle for permet de perdre du temps en faisant appel à la fonction wait().
La ligne 15 allume la led. La ligne 19 l'éteint.

Organigramme de ce programme (issu du rapport de M. Sunier):


2 chargement des définitions du processeur
3 declaration des E/S
4 déclaration de la fonction wait
6 déclaration des variables locales
7 initialisation de P2.8 en sortie
9 P2.8 = 0 (éteint la LED)
150'000 x (var. i)
10 appel de la fonction wait
8 infiniment (while(1))
11 P2.8 = 1 (allume la LED)
150'000 x (var. i)
12 appel de la fonction wait

Cet exemple se veut didactique et m'a permis de prendre connaissance vis à vis de la création d'un projet, de sa configuration et de sa compilation.
Il faut retenir ici qu'il y a 2 méthodes pour tester ce programme:

- soit on utilise CrossView pour le tester en mode pas à pas et dans ce cas il faut bien cocher les options dans Tasking,
- soit on utilise FlashTools pour flasher la mémoire.

(Il faut veiller à bien choisir les options de configuration: plage d'adresse, format de sortie du fichier compilé pour ces 2 méthodes).

7. L'électronique

Dans ce chapitre il sera question de toute la partie électronique du projet. C'est cette partie sur laquelle le plus de temps a été passé, et c'est celle-ci qui dans des prochaines versions de ce projet devrait être améliorées.
Une étude sur la source infrarouge et sur le capteur optique est d'abord réalisée. Puis le traitement du signal issu d'une photodiode est décrit, tout comme l'électronique qui pilote le moteur à courant continu.

7.1. Etude sur la source infrarouge

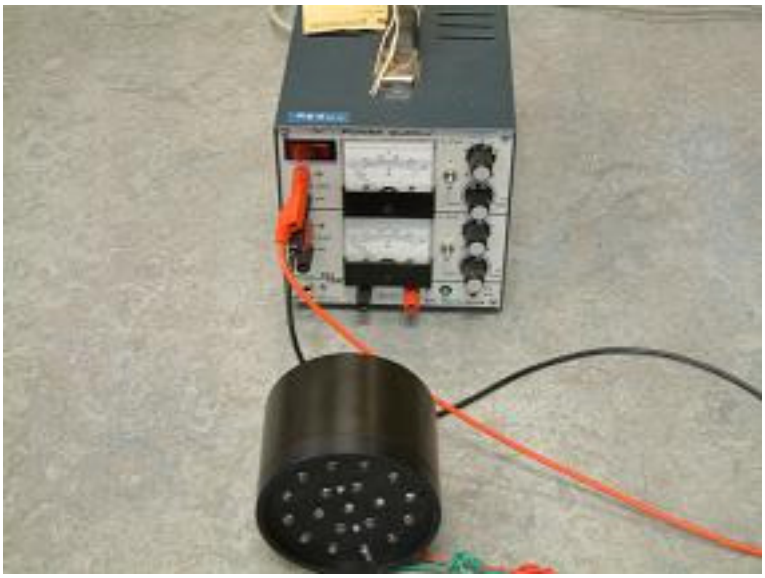


Figure 5: Source infrarouge et alimentation +12V

Cette source est composée d'une led rouge qui sert de témoin de fonctionnement. On alimente la source infrarouge directement avec une alimentation continue de +12v environ. (En fait ni mode d'emploi, ni document n'était fourni avec cette source, j'ai dû aller au laboratoire d'optique (au sous-sol) pour réaliser certaines mesures avec cette source. La personne en charge de ce laboratoire m'a conseillé d'alimenter cette source à +12V environ).

Caractéristiques des leds infrarouges:

HSDL 4230 (référence de chez Distrelec : 630420)
Angle de rayonnement : 17°
Longueur d'onde : 875nm
Intensité radian (Strahlstärke): $I_{E=}$ 150 mW/sr (I_F=100 mA)
Grande puissance de sortie
Vitesse élevée
Temps de montée 40ns
Courant nominal 100 mA
Courant de crête max. 500 mA

Nombre de led infrarouge :

19 (les leds sont inclinées différemment les unes des autres).

Portée du signal infrarouge:

La puissance par secteur « irradié » par unité (irradiance EI) diminue avec le carré de la distance de la source r [m] depuis la source. Ainsi la relation entre l'irradiance et l'intensité donne :

$$EI = \frac{I_E}{r^2}$$

Nous pouvons calculer la distance maximum théorique à partir de la relation précédente

$$r_t = \sqrt{\frac{I_E}{EI_{min}}}$$

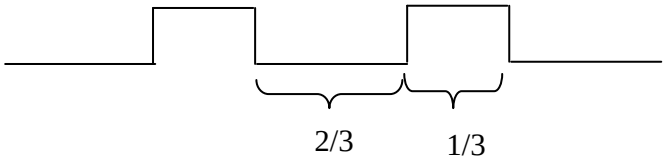
Avec I_E = 150 mW/sr et EI_{min} =

(Avec la partie pré amplification du circuit, qui sera vue plus loin, il fut possible de recevoir le signal de cette source à environ 80-100m, le signal reçu à une tension crête à crête de 30mV et 10mV crête à crête de bruit, il restait donc une marge avant que ce signal ne soit totalement noyé dans le bruit).

Modulation employée:

ON/OFF

Rapport cyclique: 2/3



Allure du signal (signal du dessus : non amplifié, signal du dessous : amplifié):

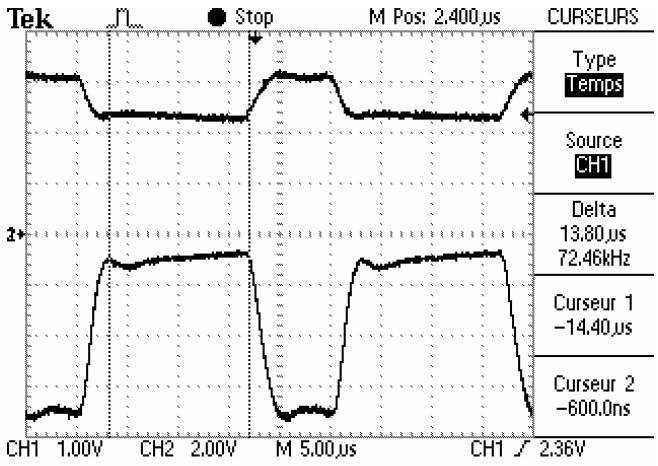


Figure 6: Allure du signal modulé reçu par la photodiode

La source comporte un circuit réalisant cette modulation et ce à la fréquence de 40kHz. Il faudra donc prévoir un filtre passe-bande à 40kHz par la suite afin de ne recevoir les signaux infrarouges seulement issus de cette source.

S'il venait à y avoir des conflits au niveau des fréquences, il faudrait modifier la fréquence dans cette source. En effet durant le projet et selon la position des câbles du circuit, le montage (fils) faisait antenne et recevait un signal qui émettait justement à la fréquence de 40kHz !

7.2. Etude sur le capteur optique

Voici d'abord l'allure de ce capteur dans lequel est inséré un module 4 quadrants

Figure 7: Capteur optique

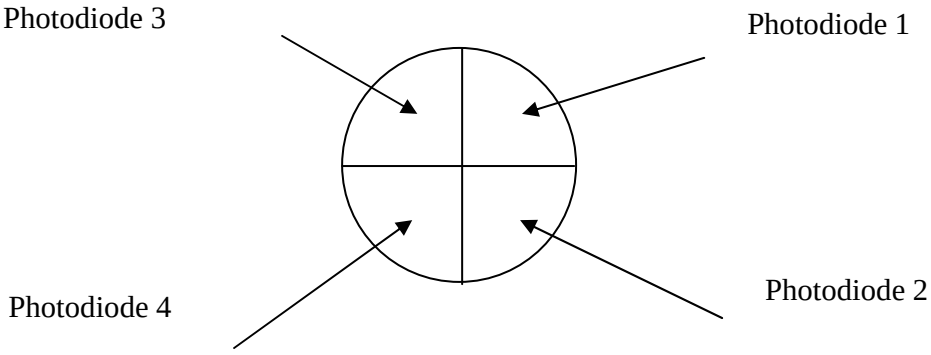
Le capteur optique (réalisé au laboratoire d'optique) est conçu avec 4 photodiodes (regroupé en un seul module).

Caractéristique de ce module 4 quadrants:

Fabricant : UDT
Série : Spot 9D (4 quadrants)

Figure 8: Module 4 quadrants

Ce capteur est idéal pour les applications de centrage de position.
ns de temps de montée.
La convention suivante est donnée pour les numéros de photodiodes:



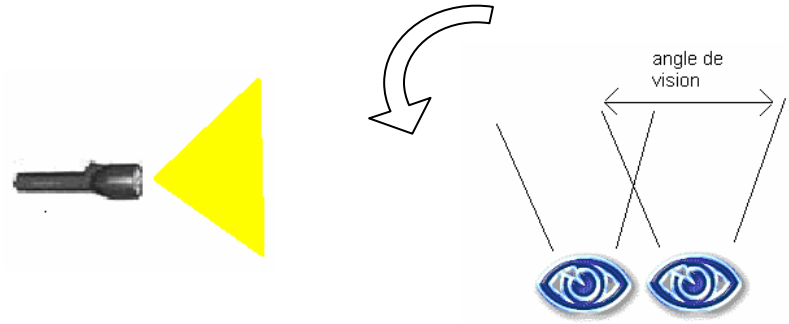
Pourquoi 4 photodiodes ?

Le fait d'avoir 4 photodiodes permet, comme avec nos yeux, de savoir si l'on se rapproche ou pas de la cible et de se centrer exactement sur la cible.

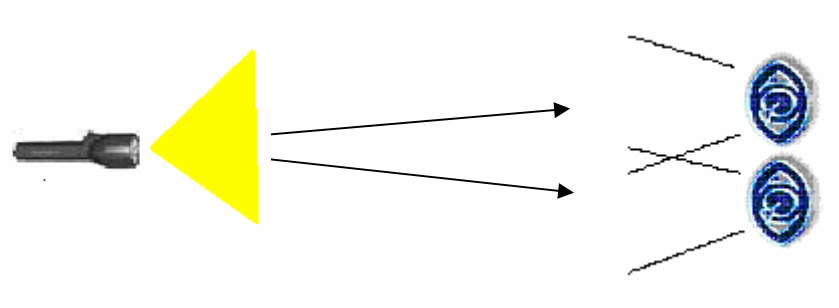
L'exemple suivant décrit les étapes pour aligner notre regard sur une source lumineuse (issue d'une lampe torche).

Au 1er instant, ne sachant pas où se trouve la source lumineuse, nous tournons la tête (ici dans le sens anti-horaire) : nous allons d'abord vite tourner la tête, puis une fois que l'œil gauche va commencer à distinguer la source nous allons tourner la tête moins rapidement, et ce de plus en plus lentement jusqu'à ce que notre tête et notre regard soit centré (instant 2).

Instant 1:



Instant 2



La lampe torche représente la source infrarouge, et les 2 yeux représentent les photodiodes du dessus ou du bas du capteur 4 quadrants (1 et 3, ou 4 et 2).

C'est cette méthode qui sera utilisée pour aligner notre capteur optique selon l'azimut. On peut voir à la figure suivante une vue de dessus du capteur optique, les flèches indiquent les 2 sens de rotation dans lequel se déplacera le capteur (grâce au moteur azimut).

Centre de l'axe de rotation du moteur « azimut »

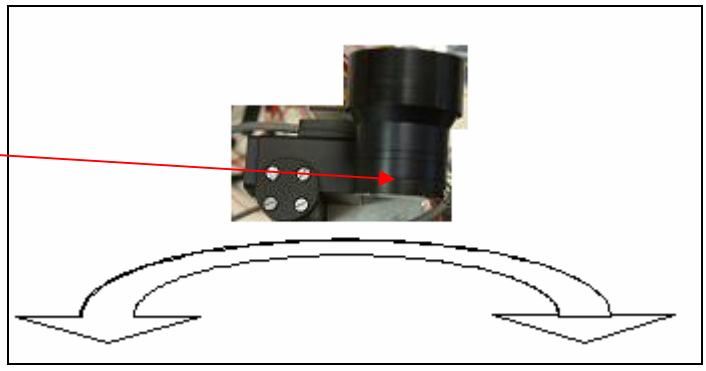


Figure 9: Vue de dessus du capteur optique et positionnement selon l'azimut

Cet exemple montrait la méthode pour se positionner selon l'azimut ; pour se positionner selon l'élévation diverses possibilités existent : comme utiliser les photodiodes 3 et 4 ou 1 et 2.

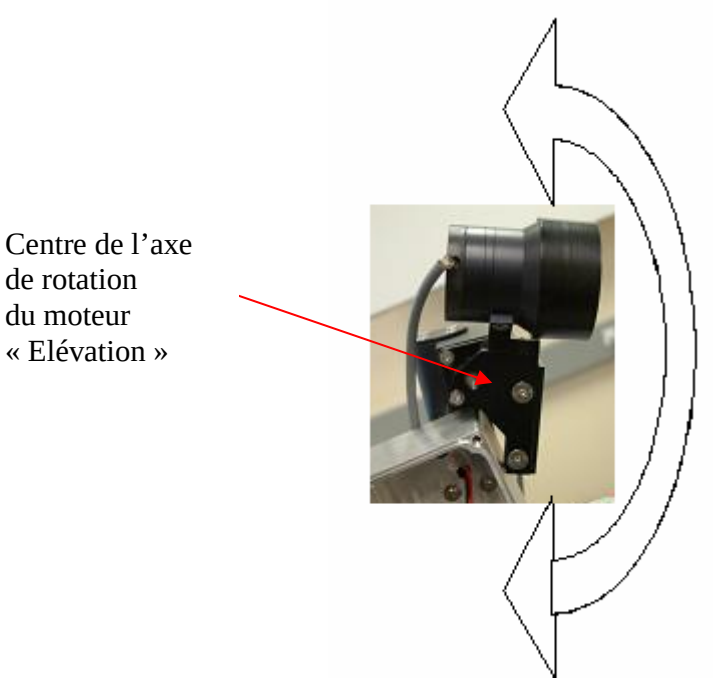


Figure 10: Vue de coté du capteur optique et positionnement selon l'élévation

7.2.1. Les photodiodes

4 photodiodes composent le module 4 quadrants. Comment fonctionnent-elles ?
Une photodiode est un semi-conducteur formé par une simple jonction P-N

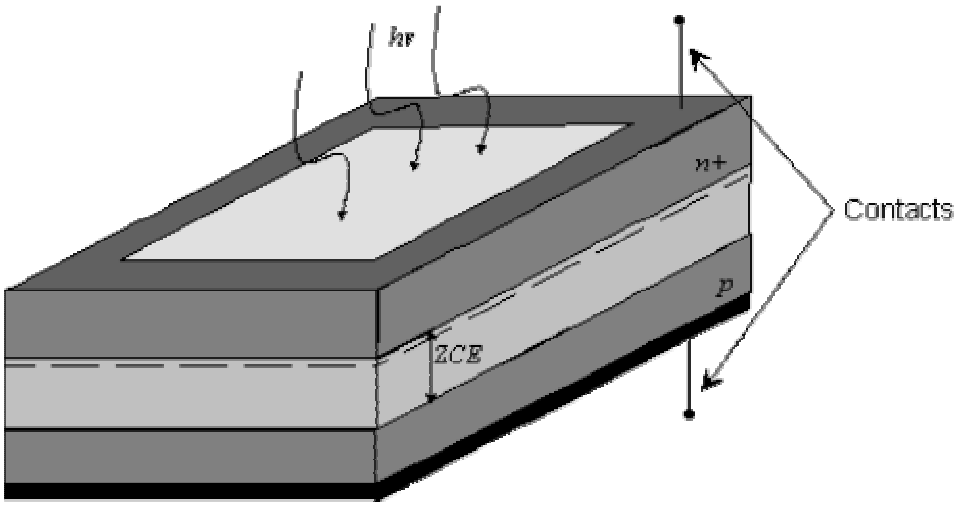


Figure 11:Structure d'une photodiode

On trouve les photodiodes dans la détection rapide des signaux faibles, asservissement, photométrie, instrument de mesure, lecteurs de code barre, spectrophotométrie, télécommunication par fibre optique, imagerie médicale.

Caractéristiques électriques d'une photodiode :
Une photodiode peut être représentée par une source de courant Iph (**une source de courant pour la simple raison que le courant que fournit la photodiode dépend uniquement de l'intensité de lumière**). En parallèle avec la source, on trouve la capacité de jonction et une résistance de shunt Rsh d'une valeur élevée et qui décrit la fuite de courant. Ensuite il existe une résistance interne de la photodiode appelé Rs et qui est en série.

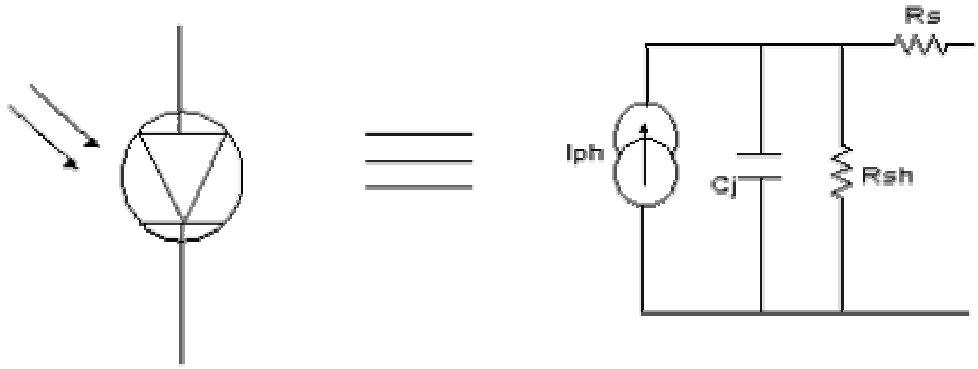


Figure 12: Symbole d'une photodiode (gauche) et son circuit équivalent (droite)

Non linéarité : une photodiode est linéaire (courant généré proportionnel à la puissance optique émise) du moment où on ne dépasse pas un certain niveau d'éclairage qui peut faire saturer le courant généré par la photodiode. Si on dépasse ce seuil, la photodiode devient non linaire c'est à dire que le photocourant n'est plus proportionnel avec l'intensité de la lumière.

7.2.2. Montage de base pour amplifier les signaux issus de photodiode

Circuits souvent associés aux photodiodes :
Le courant issu d'une photodiode est très faible, il varie de quelques pico ampères à quelques micro-ampères selon l'intensité de la lumière et la surface de la photodiode.
La première étape consiste à amplifier le courant. (Voir la partie pré amplification de ce rapport).
Cette partie théorique est issue d'Internet, elle n'a été mise en œuvre qu'en partie (le montage réalisé en pratique est étudié à la partie pré amplification).

Pour obtenir un temps de réponse qui se rapproche de la constante du temps du photo détecteur, il faut donc que le photo courant soit converti en tension sans que le temps de réponse du détecteur soit affecté. L'utilisation d'un montage amplificateur transimpédance à amplificateur opérationnel (A.O.P) est un montage souvent utilisé, d'une part parce que la résistance de contre-réaction employée pour la conversion courant tension n'affecte pas le temps de réponse du capteur (l'ampli a une très large bande passante), et d'autre par parce que le bruit généré par cette résistance est divisé par le gain de l'ampli.

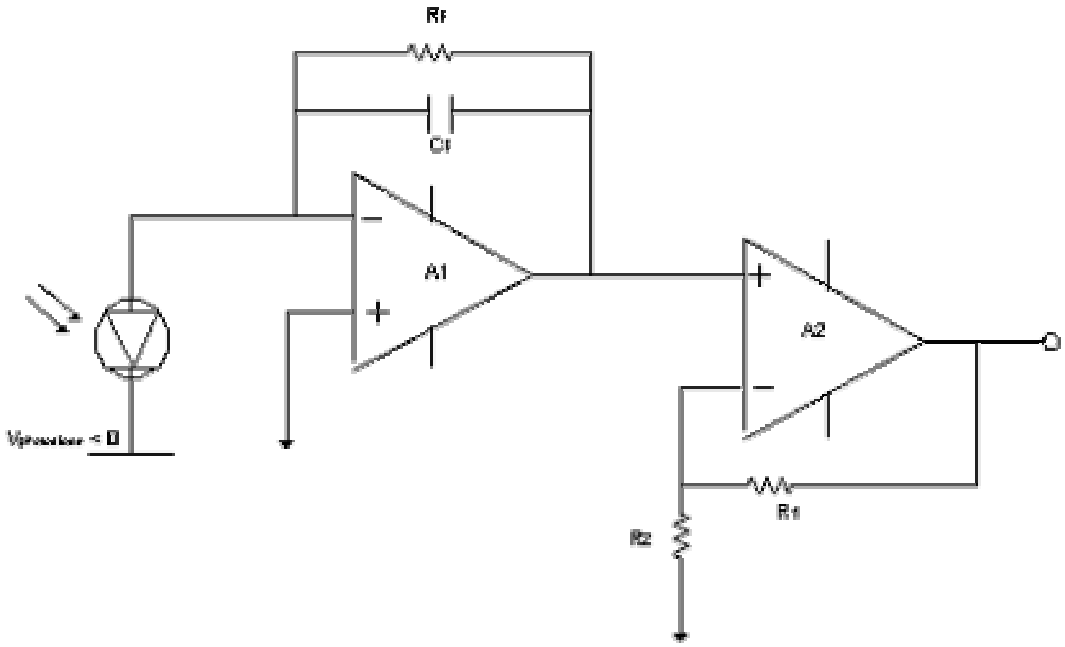


Figure 13: Amplification du signal d'une photodiode

L'amplificateur opérationnel utilisé doit avoir une très large bande passante, des courants d'entrée très faibles, et par la suite une impédance d'entrée très grande, et un bruit très faible. Typique l 'AOP doit avoir un GBW > 200 Mhz et des courants d'entrés de l'ampli ne dépassant pas quelques centaines de femtoAmpères (10^{-15} A).
Ces AOP sont chers, M. Sandoz m'a conseillé un AOP LM 6142 (celui-ci a avec un GBW de 17 MHz, 170nA de courant d'entrée et un bruit (à f=1kHz) de 16 $\frac{nV}{\sqrt{Hz}}$ et de 0.22 $\frac{pA}{\sqrt{Hz}}$)
Ces AOP sont disponibles au prix de 10.80CHF (avec 2 AOP par boîtier).

7.3. Traitement du signal issu d'une photodiode

Cette partie traite des différents montages réalisés en pratique, et qui sont utilisés pour le fonctionnement du projet. La stratégie de fonctionnement du projet dans sa globalité y est décrite de façon claire et simplifiée, puis viendra des sous parties décrivant chaque circuit.

7.3.1. Stratégie de traitement du signal

Pour aligner le capteur optique en face de la source infrarouge (en azimut) il est nécessaire de traiter les signaux provenant des photodiodes. Ici le principe du traitement du signal **pour une seule photodiode** est décrit.

Les photodiodes émettent du courant, grâce au montage de pré amplification c'est sur les tensions alternatives que le traitement aura lieu.

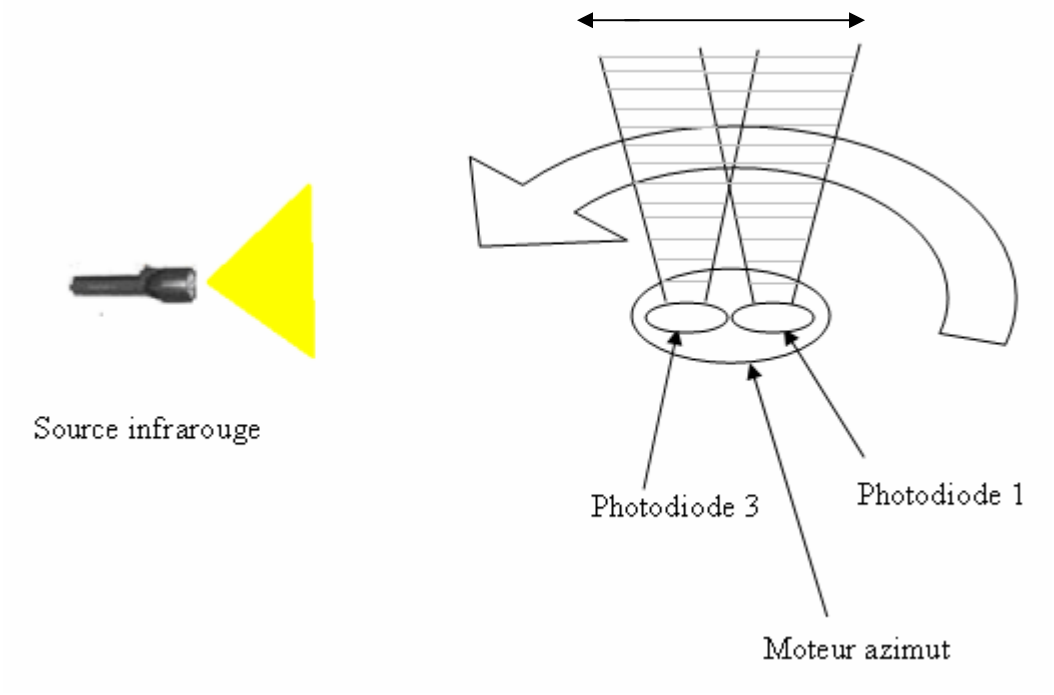
Le but du montage est de lire la tension continue provenant d'une photodiode tous les certains temps.

Pourquoi lire une tension continue ?
Car le microcontrôleur C167CR dispose d'entrées analogiques, et un convertisseur analogique numérique permet de lire cette tension continue. On doit donc lire une tension continue et pas alternative.

Si l'on se réfère à l'exemple 7.2, plus la photodiode sera en face de la source infrarouge et plus sa tension augmentera. Quand ce niveau de tension sera le plus élevé c'est que la photodiode sera en face de cette source.
L'utilisation de 2 photodiodes permettra de faire tourner le moteur (azimut) **dans le bon sens**.
En effet si la photodiode 3 délivre un courant plus important que la photodiode 1 c'est qu'il faut tourner dans le sens anti-horaire.

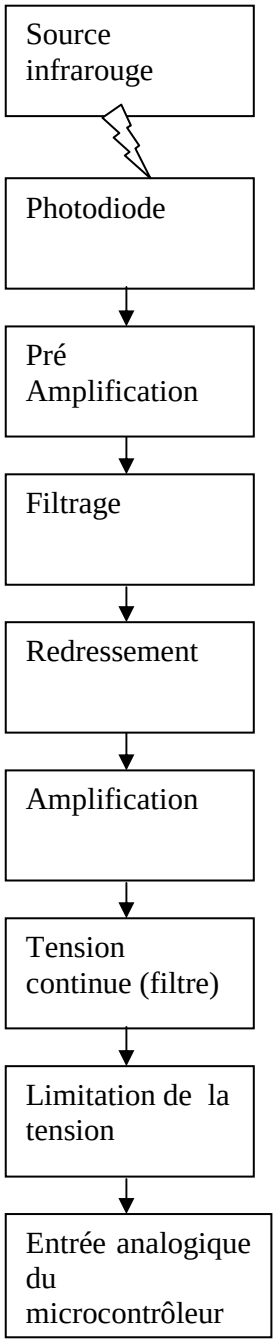
Si l'on regarde la figure suivante :
Au début aucune des 2 photodiodes ne voit la source infrarouge.
Le programme devra donc faire tourner le moteur azimut dans un certain sens (admettons le sens anti-horaire pour cet exemple). La photodiode 3 va commencer à détecter la source.
Un test dans le programme devra réalisé cette opération :
Si le niveau de tension de la photodiode 3 est supérieure à celui de la photodiode 1 alors tourne dans le sens anti-horaire.
Quand ces 2 niveaux de tension sont les mêmes alors arrête le moteur.

Vision des 2 photodiodes



Comme vu auparavant on sait que la photodiode émet du courant selon la longueur d'onde quelle reçoit.
La photodiode utilisée est sensible au domaine de l'infrarouge, ce domaine est présent presque partout mais avec des niveaux différents de puissance. De plus, des sources infrarouges (comme les pointeurs utilisés lors des conférences) sont facilement disponibles dans le commerce. Quelqu'un qui s'amuserait à pointer son faisceau infrarouge en face de ce capteur pourrait fausser le positionnement. Les concepteurs de la source infrarouge ont donc pensé à implémenter une modulation à 40kHz. Le montage qui recevra ce signal devra donc être sensible non pas à la composant continue du signal mais au signal modulé à 40 kHz.
Ainsi le circuit traitant ce signal modulé doit d'abord comporter une pré amplification du signal provenant de la photodiode. Ensuite ce signal est filtré pour ne garder « que » le signal de 40kHz. Un redressement intervient alors, le signal est à nouveau amplifié du fait que l'amplitude baisse durant le filtrage et le redressement.
Un filtre permet enfin d'obtenir la tension continue. Finalement il faut veiller à ce que la tension délivrée soit compatible avec l'entrée TTL du microcontrôleur (0 5V), il faut donc limiter la tension qui arrive sur l'entrée analogique.

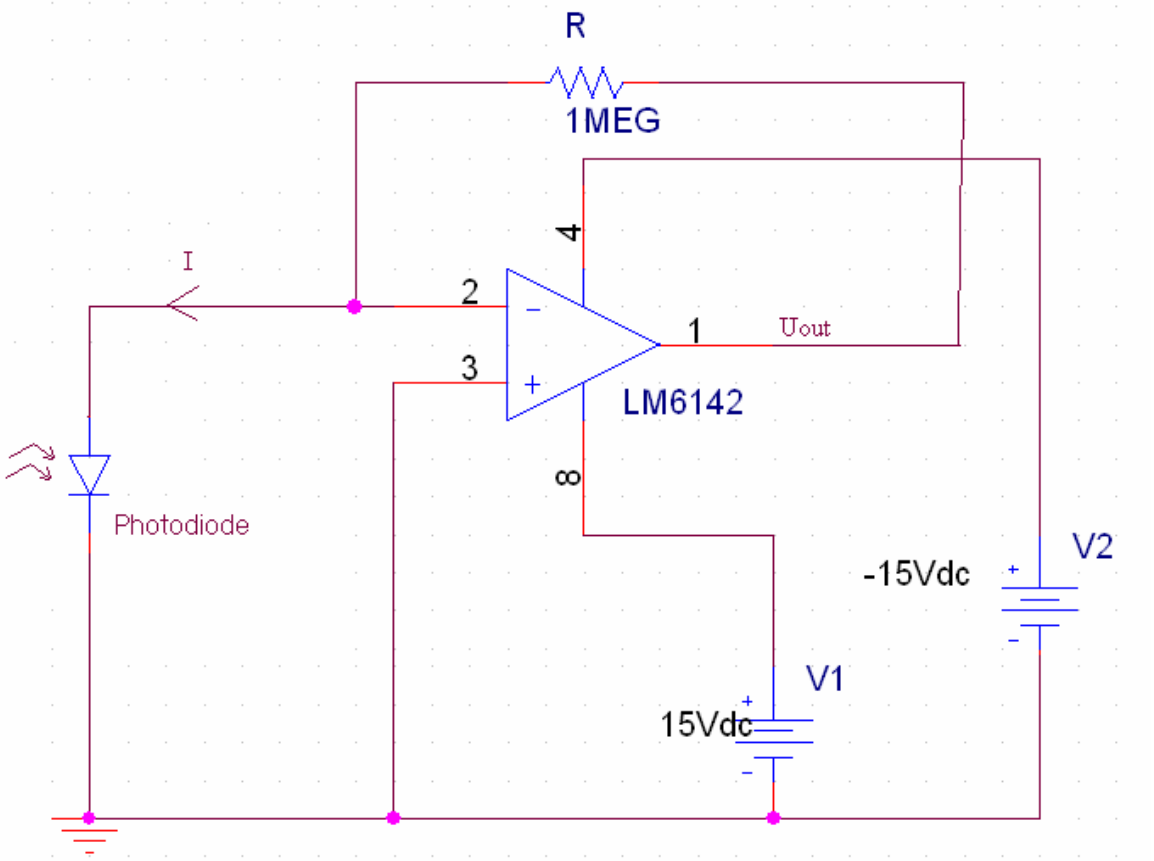
Voici le résumé des opérations pour traiter le signal provenant d'une photodiode. Les mêmes opérations sont à effectuer autant de fois qu'il y a de photodiode.



7.3.2. La pré amplification

Les signaux issus des photodiodes ne sont pas des tensions mais des courants très faibles, il n'est donc pas possible de travailler directement sur ces courants. Un AOP courant tension permet de travailler dans le domaine des tensions et d'amplifier le signal.

Schéma du circuit :

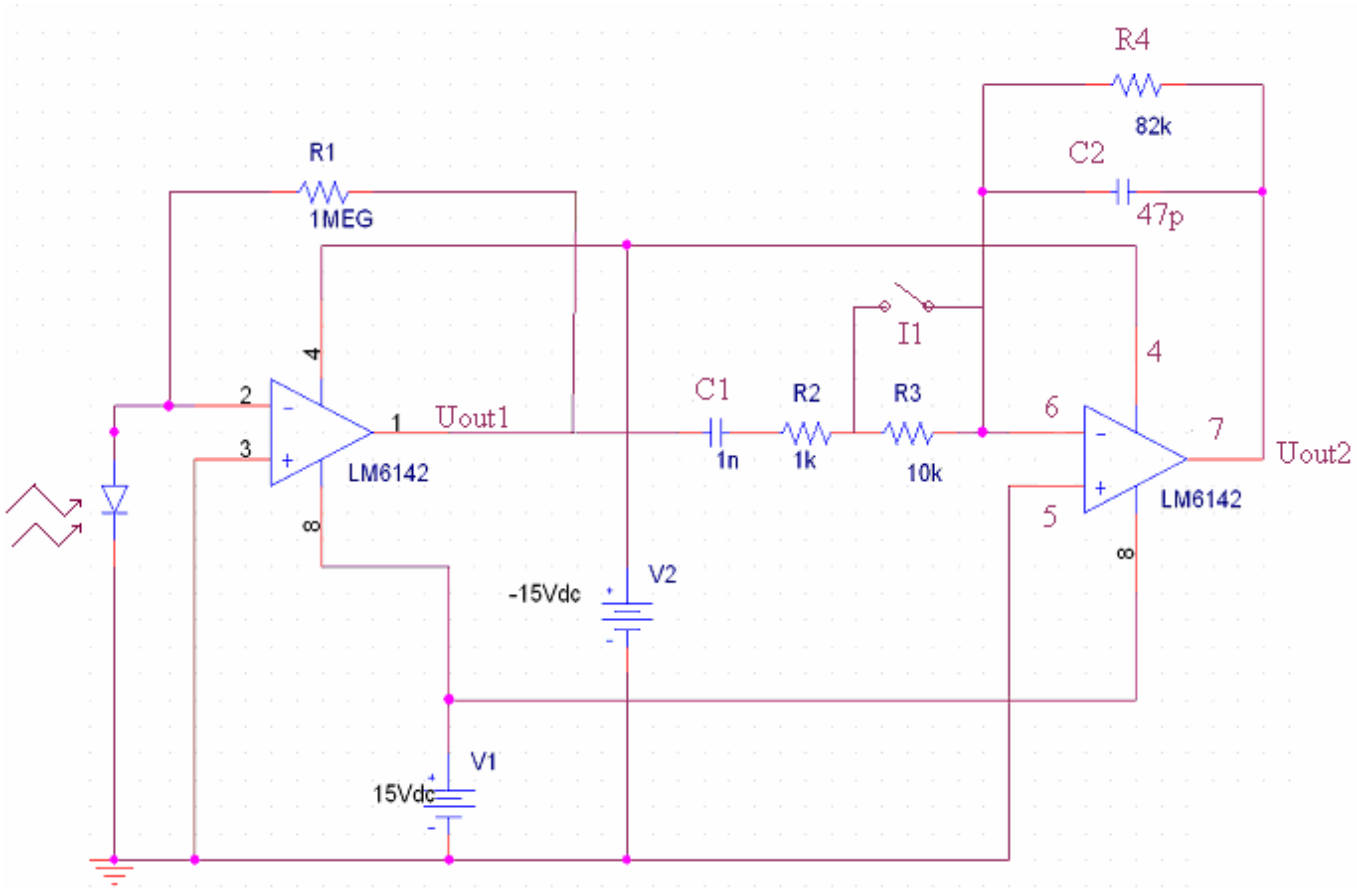


(R est choisi grand pour avoir une grande tension).
A la sortie de l'AOP (patte 1) : on a Uout= -R* I.

Fonctionnement sans la source infrarouge :
Si l'on déplace la photodiode on voit que le niveau de tension continue bouge.

Fonctionnement avec la source infrarouge :
Plus on met la photodiode en face de la source et plus la tension crête à crête augmente.

Au niveau de la distance :
En plaçant la photodiode proche de la source infrarouge la tension alternative délivrée sur Uout sature à $^{+}_{-}$15V . Plus on s'éloigne de la source et plus la tension alternative sur Uout diminue. (La diminution est d'autant plus rapide que l'on s'éloigne du fait qu'elle baisse selon le carré de la distance).
Avec ce montage on perçoit le signal jusqu'à quelques mètres.
Un 2ème ampli est utilisé pour augmenter la distance de réception : on passe de quelques mètres à plusieurs dizaines de mètres.



Un cavalier (ici représenté par l'interrupteur I1) peut être placé pour modifier le gain (*82 ou *7)
Si l'interrupteur I1 est fermé :
A la sortie de Uout1 on branche un 2ème AOP avec un gain de Gain= - R4/R2 = 82

Interrupteur I1 ouvert:
Le gain passe à -R4/ (R2+R3)= 7.45

Utilité du condensateur C1 :
Il coupe la composante continue provenant des infrarouges présents.

Utilité du condensateur C2 :

Avec R4 et C2 on a un filtre passe-bas de 41.3kHz (fc= $\frac{1}{2*\pi*R_4*C}$)

Le condensateur C2 empêche que le montage entre en oscillation. La fréquence de ce filtre est prise au-dessus des 40 kHz, il ne faudrait pas diminuer la tension du signal modulé.

Avec ce montage il a été possible de placer la source à environ 80-100m de distance et de recevoir le signal en Uout2 avec 30mV crête à crête et 10mV crête à crête de bruit ! Ce résultat est l'œuvre de nombreux essais et de nombreuses configurations de montage. Il a été réalisé dans les couloirs de l'EICN au rez-de-chaussée.
Cette distance peut bien sûre encore être augmentée, une partie « amélioration » est présentée plus loin dans ce rapport mais n'a pas été réalisé pratiquement..

7.3.2.1. Simulation Pspice

Ce projet a été l'occasion de tester le logiciel Pspice. Le montage précédent a été simulé (la photodiode a été remplacée par une source de courant Ipulse, je ne savais pas s'il était possible de simuler des photodiodes).
Comme de nombreuses simulations ont été réalisées, il n'aurait pas été envisageable de toutes les faire figurer dans ce rapport. Une simulation Pspice de la partie pré amplification est disponible à la partie 7.3.10 « amélioration » cette partie sera plus approfondie.

7.3.3. Le filtrage

Maintenant que la pré amplification est réalisée, il faut filtrer le signal pour ne garder que la fréquence de 40kHz. (Le schéma de ce filtre est présenté dans la partie PCAD)
Différents montages ont été réalisés, M. Sandoz m'a proposé un montage analogique:
Ce montage a une fréquence de résonance à 40kHz. Le premier potentiomètre permet de régler la bande passante, et le second de régler la fréquence de résonance.
L'impédance d'entrée vaut $Zin= j*\omega*C*R^2$ soit une inductance équivalente de valeur $L_{équiv}=CR^2$
Ce filtre présente le désavantage que plus on limite la bande passante et plus le signal de 40kHz est atténué, ce qui diminue fortement la distance de réception. Il serait utile de réaliser un filtrage numérique plutôt qu'analogique.

7.3.4. Le redressement

L'utilité du redressement est de donner plus d'énergie afin d'extraire la tension continue. (Rq: si le signal dont on voulait extraire la composante continue était sinusoïdale et que l'on ne le redressait pas, alors sa tension continue serait nulle !).
2 montages ont été testés avec et sans AOP. Le montage à pont de diodes (Pont de Graëtz) pose le problème que l'on perd les tensions de jonction des diodes soit 2*Uj=1.2v. Cette perte de tension est trop élevée et ferait perdre en efficacité le montage, c'est pourquoi le montage suivant a été retenu : montage redresseur double alternance sans seuil. (Schéma disponible à la partie PCAD)

Fonctionnement :

Un sommateur à 2 entrées est utilisé, la 1ère entrée reçoit la sortie d'un redresseur de type inverseur. La 2ème entrée dont le poids est la moitié de celui de la 1ère entrée, reçoit directement le signal d'origine, avant redressement.
Le 2ème AOP amplifie le signal (AOP inverseur).

7.3.5. Extraction de la composante continue

Le signal est maintenant redressé en double alternance et de plus il est amplifié. Un filtre est utilisé pour extraire la composante continue. Il est de type passe-bas (avec une cellule R-C).

Sa fréquence de coupure est de $\frac{1}{2*\pi*R*C} = \frac{1}{2*\pi*120k*1\mu} = 1.32\text{Hz}$

Le problème de ce filtre est qu'il est assez lent mais en pratique le moteur n'a pas une grande vitesse donc ce montage convient.
(Schéma disponible à la partie PCAD)

7.3.6. Limitation du signal

La tension continue issue du filtre précédent laisse passer une tension qui varie entre 0 et +15v maximum. (Le cas +15V arrivera si l'on place la source à quelques centimètres du capteur optique).
Le microcontrôleur dispose d'entrée analogique limitée à 0+5V (TTL). Il faut donc limiter la tension précédente. Pour cela un AOP suiveur a été utilisé, celui-ci est alimenté en 0V +5V (environ), cela limite donc le signal. La connexion sur l'entrée analogique peut enfin se faire.
(Schéma disponible à la partie PCAD)

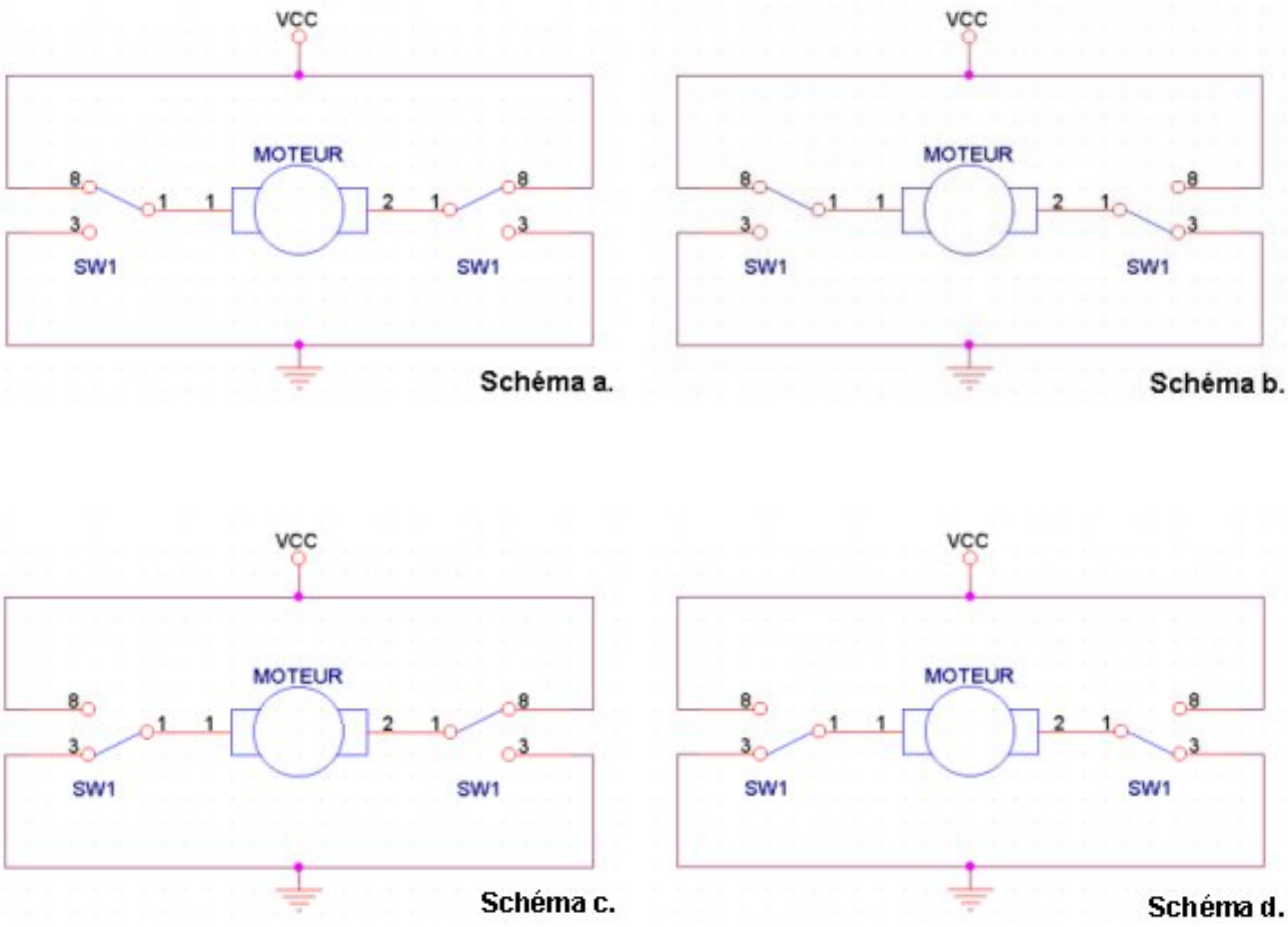
7.3.7. Pilotage du moteur azimut

Le moteur utilisé est un moteur à courant continu.
Puissance conseillée: 5W,
Tension nominale : 12V,
Courant de démarrage:2.25A.

Pour piloter ce moteur on dispose de sorties sur le microcontrôleur sur lequel on peut utiliser des unités PWM. Le moteur devra pouvoir aller dans les 2 sens. Il faut donc trouver un driver permettant de fournir assez de puissance pour le faire tourner, et de plus ce driver devra permettre de piloter le moteur dans les 2 sens il faut donc pouvoir inverser la polarité, ce driver devra permettre de varier la vitesse.

Après quelques recherches le choix s'est porté sur le driver L6203 de chez ST Microelectronics. Il s'agit d'un pont en H dont voici le principe de fonctionnement et les principales caractéristiques:
Tension max: 48V
Tension min: 12V
Courant efficace: jusqu'à 4A, (5A si t<1ms)

Principe de fonctionnement d'un pont en H:



Au schéma (a) le moteur est arrêté (on vient court-circuiter les 2 pôles, si le moteur est en marche cela revient à le freiner puis à l'arrêter).
Au schéma (b) le moteur va dans un sens (cela dépend du sens du courant),
Au schéma (c) le moteur va dans le sens inverse du schéma (b),
Au schéma (d) le moteur est à nouveau freiné puis arrêté.
Le schéma suivant montre comment est réellement le pont en H, les interrupteurs sont remplacés par des transistors.
Quand l'entrée In 1 (In 2) est active, le transistor Q1 (Q2) est passant et le transistor Q3 (Q4) est bloqué. Quand In 1 est actif, le pôle 1 du moteur est au potentiel Vcc, et lorsque In 1 est à l'état bas, le pôle 1 du moteur est à la masse.
Le tableau suivant résume les 4 possibilités de configuration du moteur:

In 1	In 2	Moteur
0	0	Arrêt
0	1	Sens 1
1	0	Sens 2
1	1	Arrêt

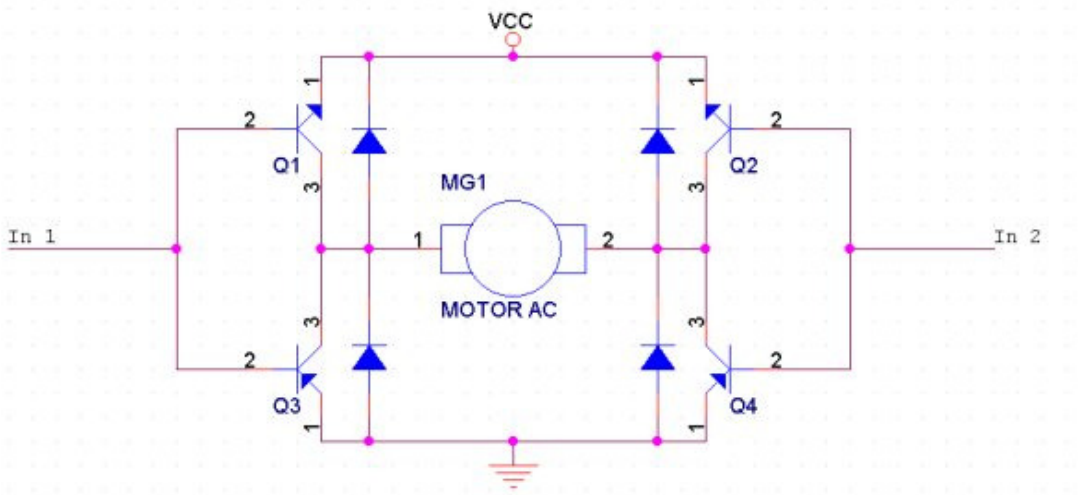


Figure 14: Pont en H

Le moteur ne s'arrête pas instantanément, il continue de tourner du fait de l'inertie (inertie du rotor du moteur azimut: 4.13gcm^2), il se comporte alors en génératrice. On utilise des diodes de roue libre pour éviter d'avoir des courants dans les transistors (ceux-ci ne sont donc pas détruits).

Comment varier la vitesse d'un tel moteur ?
Il y a 2 méthodes.
- 1ère méthode:
Elle consiste à varier la tension directement sur le moteur. Voici la caractéristique de la vitesse en fonction de la tension.

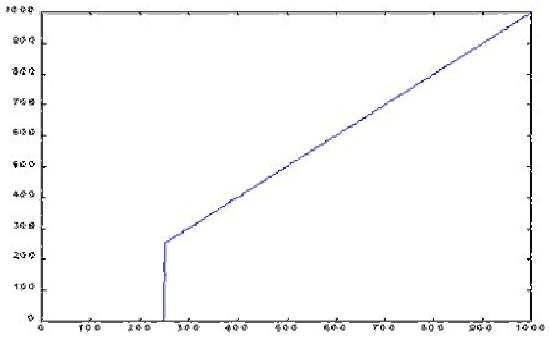


Figure 15: Vitesse d'un moteur selon la tension

On remarque qu'il faut atteindre une certaine tension pour que le moteur démarre. Lorsque la tension est trop basse, la force électromotrice ne peut pas vaincre les frottements et le rotor reste fixe. **Avec cette méthode on ne pourra donc pas avoir de petite vitesse**. Aucune régulation ne sera possible (tout du moins elle ne serait pas précise).

- 2ème méthode:
Elle consiste à utiliser la PWM (Pulse Width Modulation). Un signal PWM est un signal dont la période est fixe, mais le rapport cyclique varie. En d'autres termes, t1 et t2 varient tout en conservant t1+t2=T=constante.

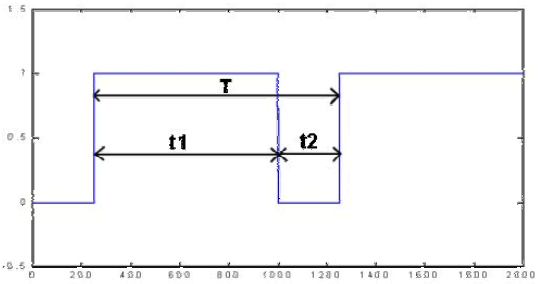


Figure 16:Allure d'un signal PWM

En utilisant la PWM on peut faire varier lentement la vitesse du moteur, mais pourquoi ?
Car pendant t1 le moteur a une tension Vcc qui permet de vaincre la force électromotrice et permet de faire tourner le moteur. Cependant la tension moyenne dépend du rapport cyclique et ce proportionnellement selon la relation suivante:

$$Vmoy = \frac{t1.Vcc}{T}$$

Il faut remarquer que si l'on choisit une période T donc une fréquence de commande dans les fréquences audibles par l'homme, on entendra un son plus ou moins désagréable. Il est conseillé d'utiliser une fréquence supérieure à 20kHz

7.3.7.1. Le driver

Le driver L6203 comporte un pont en H, il se pilote avec 2 signaux PWM:
Avec le 1er signal PWM on pilotera le moteur dans un sens, et selon le rapport cyclique qu'on lui appliquera, on pourra faire varier la vitesse du moteur.
Avec le 2ème signal PWM on pourra faire la même chose mais dans le sens inverse. Il faudra veiller à ne pas activer en même temps les 2 PWM, au risque de détruire le driver !
De plus si l'entrée 11 du driver est au niveau logique 1 alors le moteur pourra tourner, mais si elle est au niveau logique 0 le moteur ne tournera pas et ce même s'il y a un signal PWM.
Sur l'entrée 5 (IN1) on applique la 1ère PWM.
Sur l'entrée 7 (IN2) on applique la 2ème PWM.
Le moteur sera branché entre les bornes 3 (OUT1) et 1 (OUT2).
La partie programmation sera vue au prochain chapitre.

7.3.7.2. Le microcontrôleur en mode nomade

Une fois le fichier .hex téléchargé dans le microcontrôleur il est facile de l'utiliser en mode nomade (autonome).
Il suffit d'alimenter la patte VCC en +5V et de mettre sa patte GND à la masse.

7.3.7.3. Le montage

(Schéma disponible à la partie PCAD)

7.3.8. Circuit final

Il reste maintenant à connecter entre eux tous ces circuits. Le circuit réalisé pratiquement est disponible à la partie PCAD.
Pour piloter le moteur azimut on utilise 2 photodiodes, il faut donc traiter séparément chacun de leur signal. Ce schéma a donc été réalisé 2 fois. (Pour piloter le moteur selon l'élévation il faudrait faire encore 2 fois ce circuit).

Utilisation de la carte prototype :

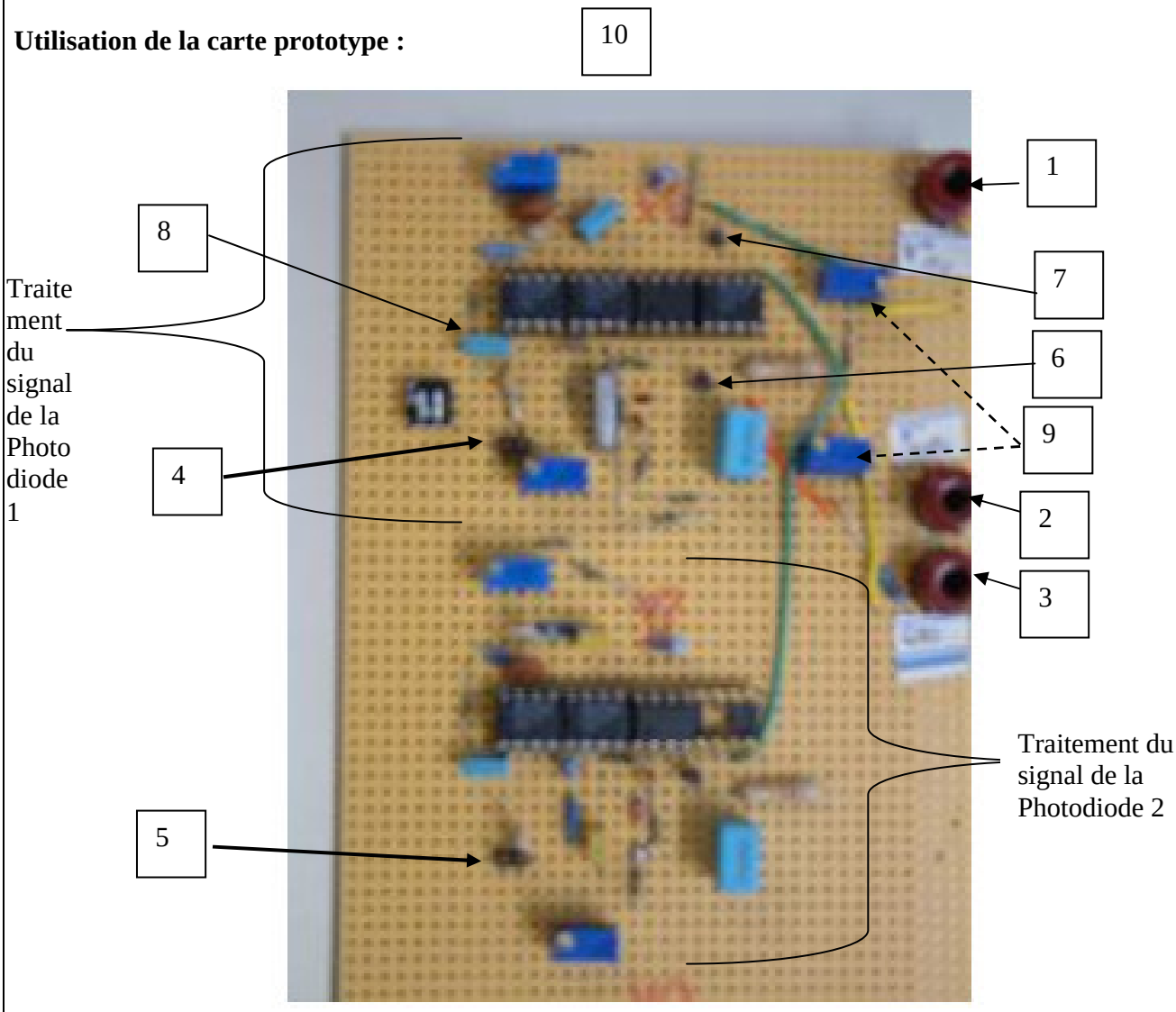


Figure 17: Carte de traitement de 2 signaux issus de 2 photodiodes

Pour l'utiliser il faut l'alimenter en +15V (repère 1) -15V (repère 2), et relier la masse (repère 3).
A partir du capteur optique il faut relier le fil provenant d'une photodiode sur un des connecteurs prévus à cet effet sur la carte (repère 4).
Un autre connecteur permet le branchement d'une 2ème photodiode (repère 5). Il faut relier la masse du capteur optique sur le connecteur situé à proximité du repère 4.
La sortie de la 1ère photodiode est disponible sur la carte au repère 6.
La sortie de la 2ème photodiode est disponible sur la carte au repère 7.

Un oscilloscope peut être utilisé pour vérifier le bon fonctionnement de l'ensemble. Une entrée de l'oscillo pourra être branchée sur le repère 8 à la sortie du condensateur en bougeant le capteur optique on doit retrouver le signal à 40 kHz (sans composante continue).
Sur le repère 6 ou 7 on devra voir la tension continue varier entre 0 et 5V (environ, cela dépend de la distance entre la source infrarouge et le capteur optique). Si la tension varie en dehors de cette plage il suffit d'utiliser les 2 potentiomètres (repère 9) et de vérifier la tension à l'aide d'un voltmètre.
Pour régler sur la bande passante et la fréquence centrale du signal à recevoir: faire l'ajustement sur le potentiomètre du repère 10.
Si ces opérations marchent, on pourra relier les 2 sorties sur les 2 entrées analogiques du microcontrôleur.

7.3.9. PCAD

La carte prototype marche correctement. Afin de voir comment se fait un circuit imprimé, j'ai du utiliser PCAD. La carte qui va être réalisée comportera les 2 circuits de traitement provenant des 2 photodiodes, le circuit de puissance pour piloter 1 moteur, les connecteurs pour placer le microcontrôleur.

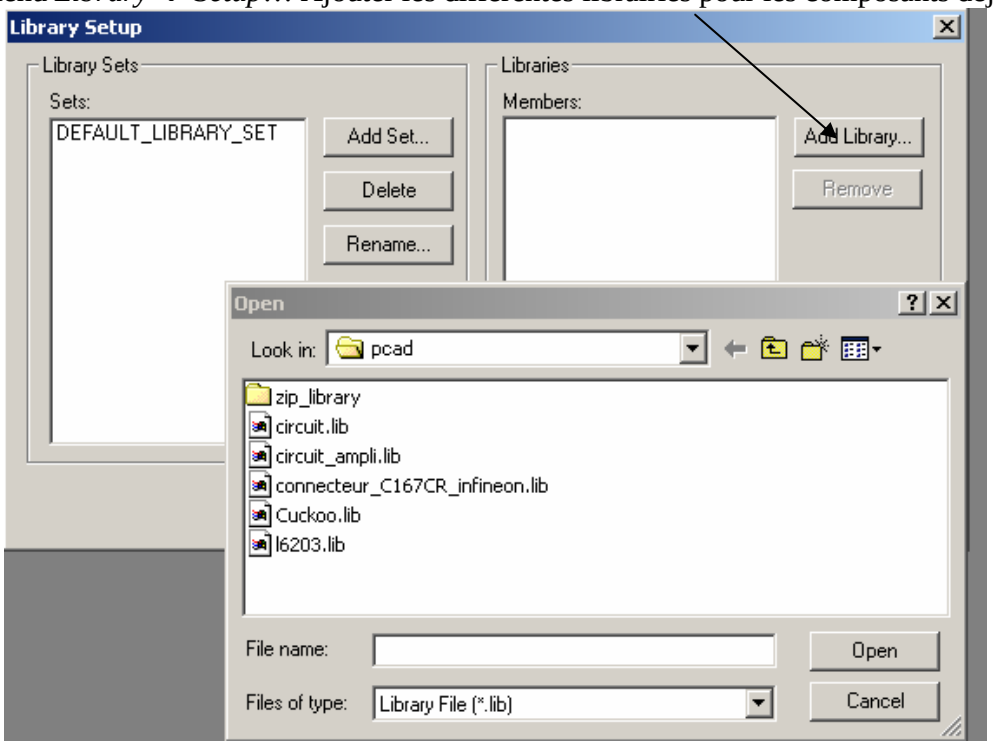
Voici la démarche pour la création de ce circuit:
- Création des librairies des composants qui seront mis sur la carte,
- Création du schéma du circuit,
- Placement des composants aux endroits voulus et routage de la carte,
- Un fichier peut être transmis à la fin à M Kiesling pour qu'il réalise le circuit imprimé.

7.3.9.1. Les librairies

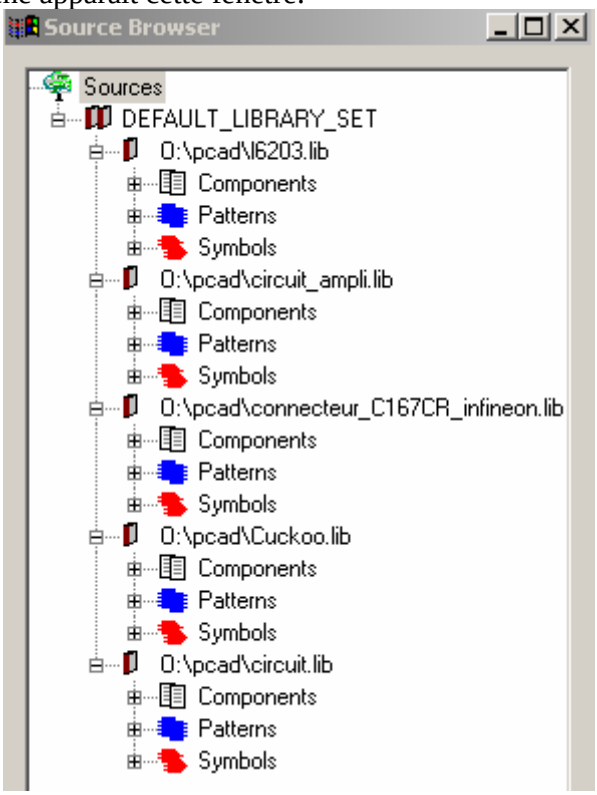
Des librairies sont déjà disponibles pour les composants tel que les résistances, condensateurs, …
Des librairies ont du être téléchargées sur Internet, et la librairie pour les connecteurs du microcontrôleur a du être réalisée.

On va dans le module suivant pour créer la librairie du projet qui regroupera tous les composants:

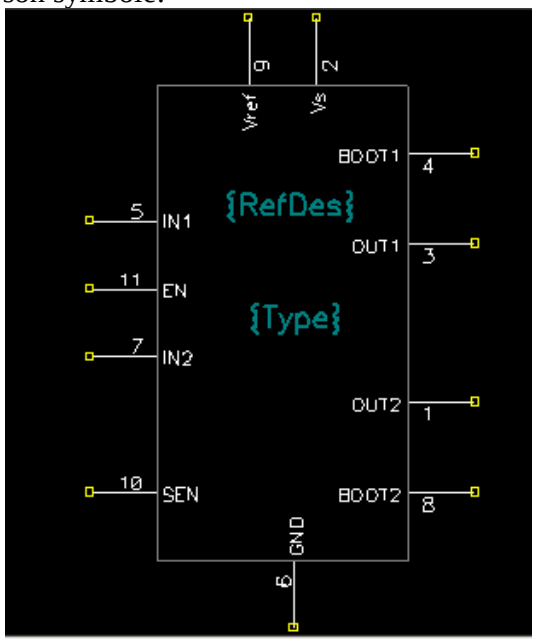
Aller dans le menu *Library* → *Setup*… Ajouter les différentes librairies pour les composants déjà existants.



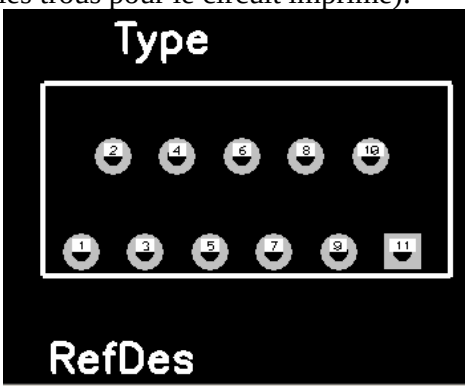
Dans le Source Browser à gauche apparaît cette fenêtre:



Dans chaque librairie on trouve: les différents composants, leur pattern et leur symbole.
Si l'on clique dans la librairie components et que l'on va chercher le L6203 on pourra voir toutes les informations de ce composant tel son symbole:



Son pattern (c'est la que seront faits les trous pour le circuit imprimé):



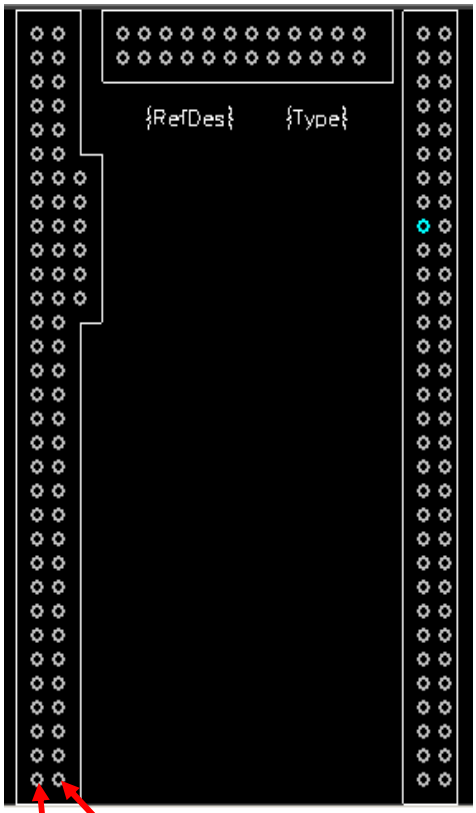
Les caractéristiques des pattes:

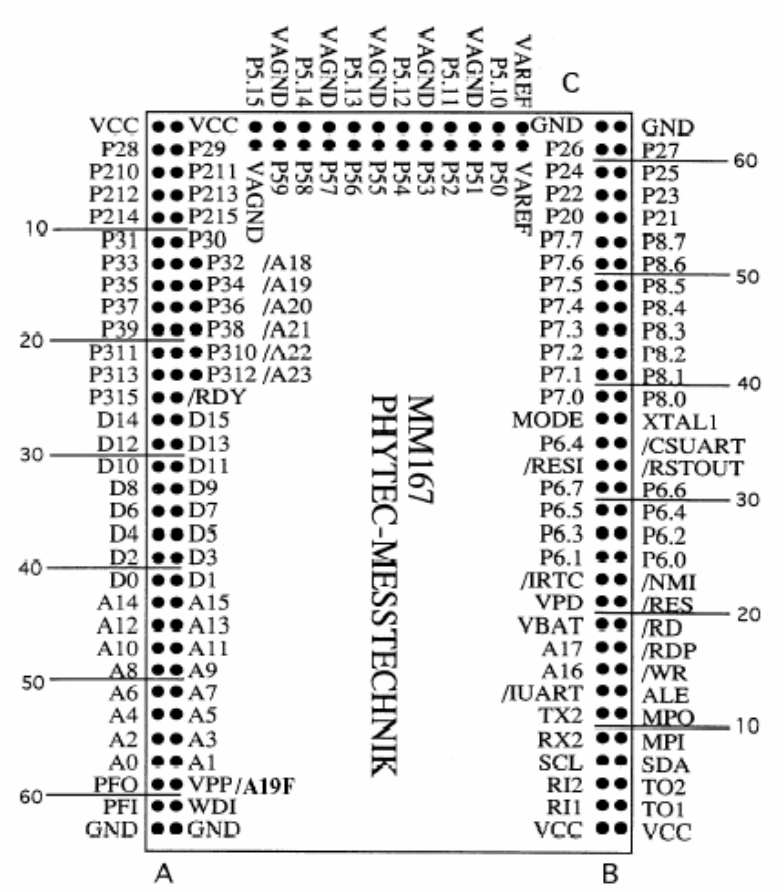
Pad # 1 1

	Pad #	Pin Des	Gate #	Sym Pin #	Pin Name	Gate Eq	Pin Eq	Elec. Type
1	1	1	1	1	OUT2	1		Output
2	2	2	1	2	Vs	1		Power
3	3	3	1	3	OUT1	1		Output
4	4	4	1	4	BOOT1	1		Unknown
5	5	5	1	5	IN1	1		Input
6	6	6	1	6	GND	1		Power
7	7	7	1	7	IN2	1		Input
8	8	8	1	8	BOOT2	1		Unknown
9	9	9	1	9	Vref	1		Input
10	10	10	1	10	SEN	1		Unknown
11	11	11	1	11	EN	1		Unknown

On trouve par exemple la patte IN1 sur lequel viendra le signal d'une unité PWM.
Les librairies utilisées pour le projet seront remises sur CD.

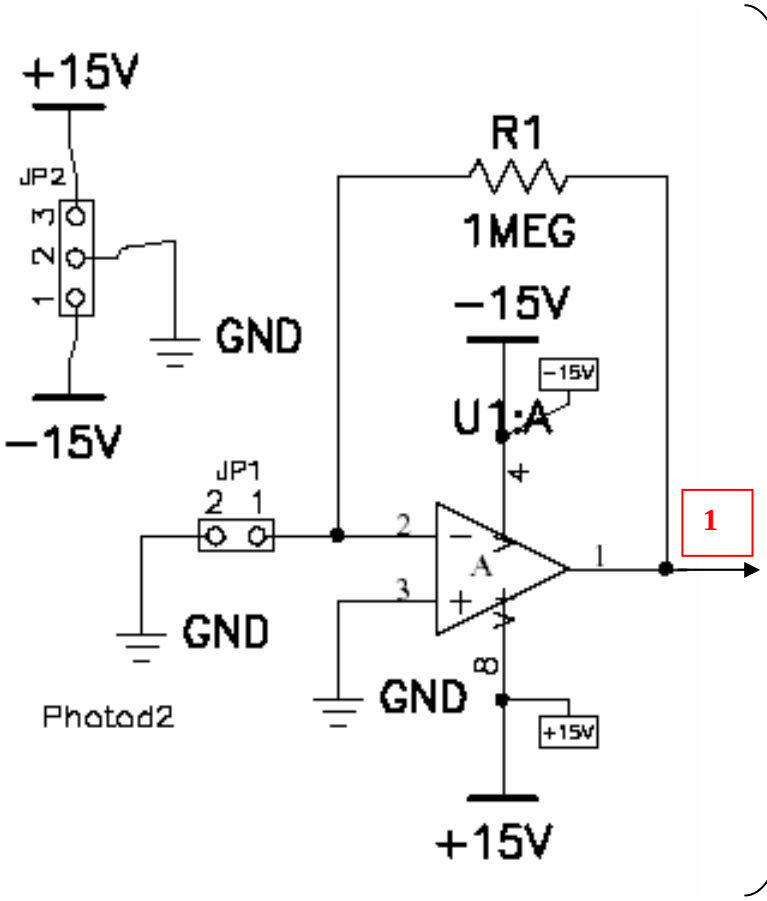
Vue du pattern créé pour mettre le C167CR:




MM167
PHYTEC-MESSTECHNIK
A
B
C
VCC P28 P210 P212 P214 P31 P33 P35 P37 P39 P311 P313 P315 D14 D12 D10 D8 D6 D4 D2 D0 A14 A12 A10 A8 A6 A4 A2 A0 PFO PFI GND
VCC P29 P211 P213 P215 P30 P32 P34 P36 P38 P310 P312 /RDY D15 D13 D11 D9 D7 D5 D3 D1 A15 A13 A11 A9 A7 A5 A3 A1 VPP/A19F WDI GND
P5.15 VAGND P5.14 VAGND P5.13 VAGND P5.12 VAGND P5.11 VAGND P5.10 VAREF
VAGND P59 P58 P57 P56 P55 P54 P53 P52 P51 P50 VAREF
/A18 /A19 /A20 /A21 /A22 /A23
GND P26 P24 P22 P20 P7.7 P7.6 P7.5 P7.4 P7.3 P7.2 P7.1 P7.0 MODE P6.4 /RESI P6.7 P6.5 P6.3 P6.1 /IRTC VPD VBAT A17 A16 /IUART TX2 RX2 SCL RI2 RI1 VCC
GND P27 P25 P23 P21 P8.7 P8.6 P8.5 P8.4 P8.3 P8.2 P8.1 P8.0 XTAL1 /CSUART /RSTOUT P6.6 P6.4 P6.2 P6.0 /NMI /RES /RD /RDP /WR ALE MPO MPI SDA TO2 TO1 VCC
10 20 30 40 50 60
60 50 40 30 20 10

7.3.9.2. Le schéma

Une fois les librairies créées, on peut passer au composant Schématic de PCAD pour réaliser les connexions entre les divers composants.



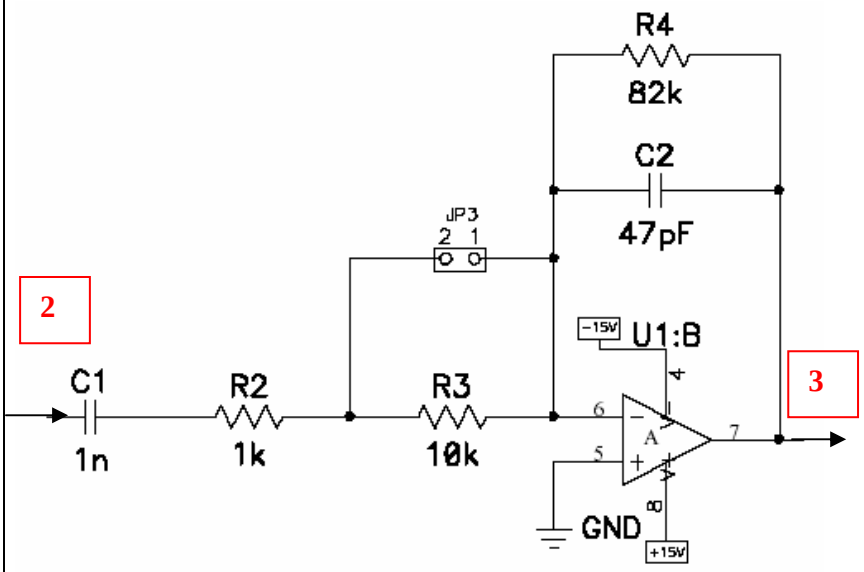
On trouve ici:

- Les bornes sur lequel seront mises les alimentations en +-15V, la masse.

- Le 1er AOP réalisant la pré amplification.

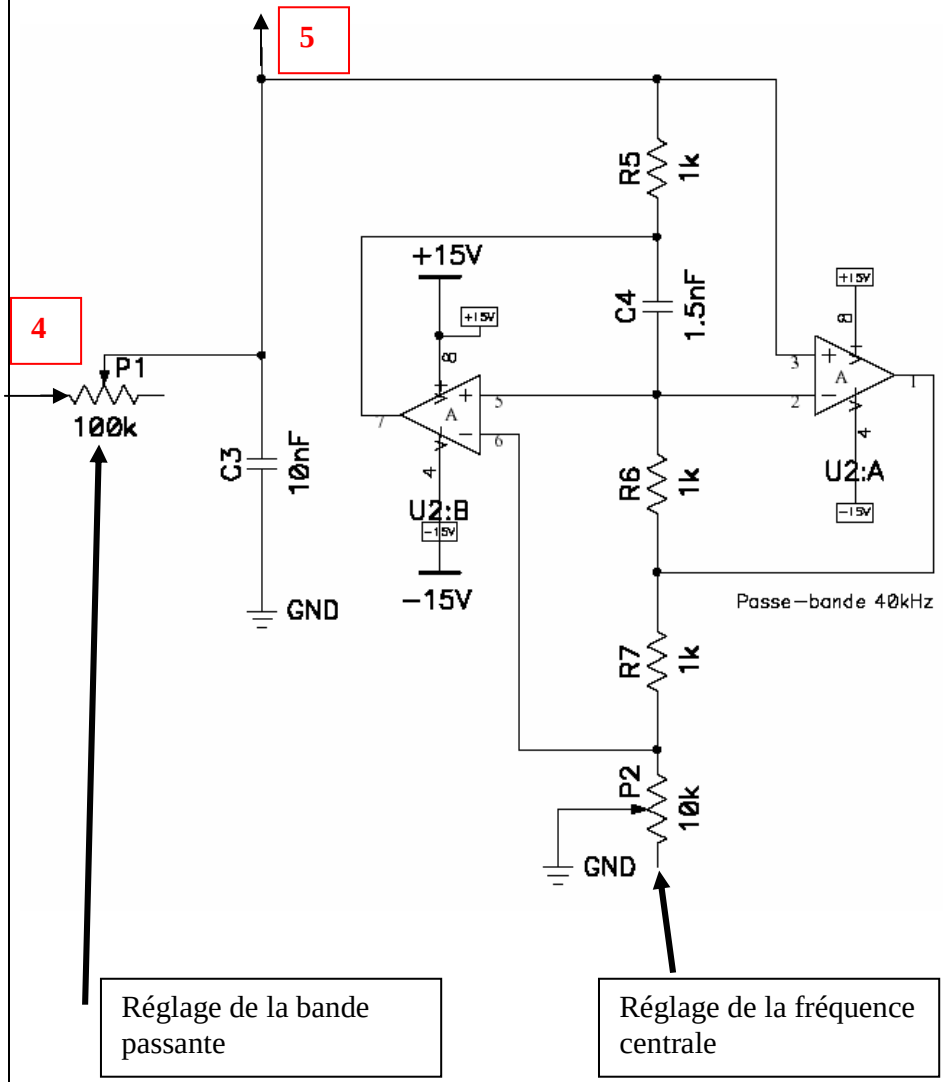
- Le connecteur sur lequel on branchera une photodiode (JP1) Sur la patte 1 de ce connecteur brancher l'anode, sur la patte 2 la cathode.

Au repère 1 on a la sortie du montage qui se dirige vers le point 2 du schéma suivant.



Amplification

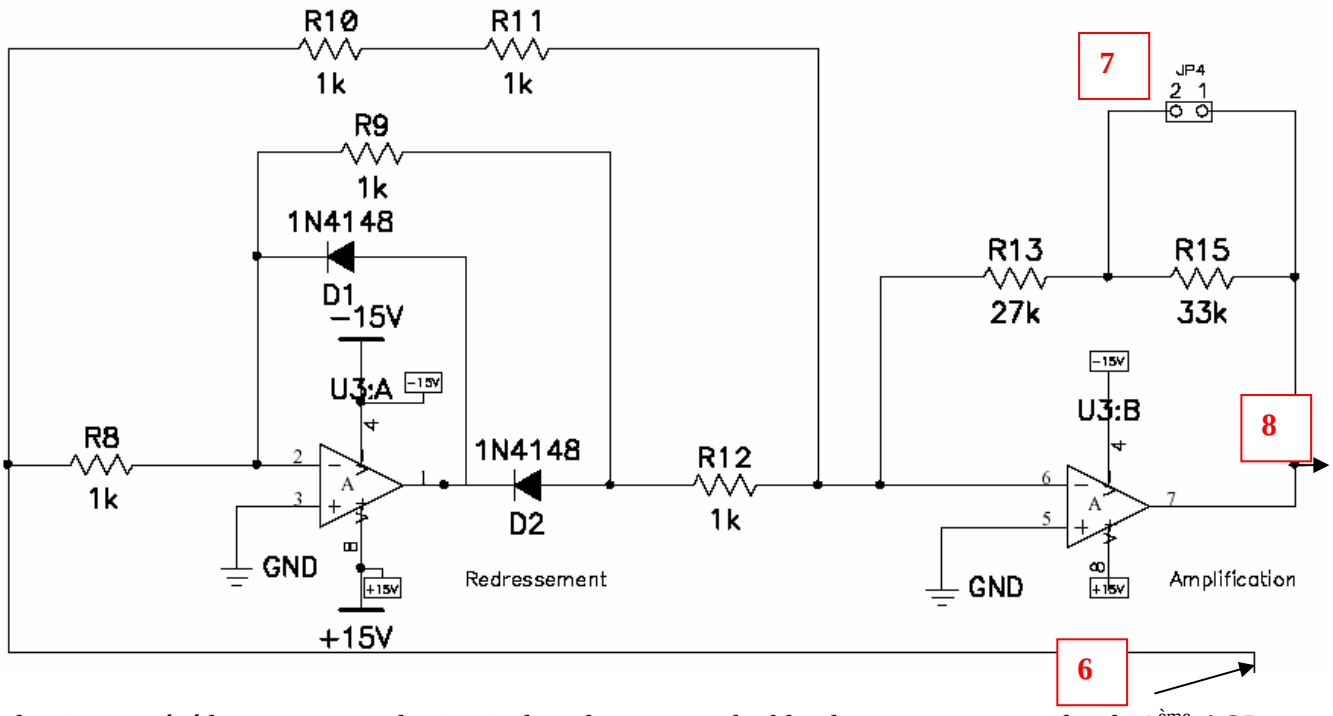
On trouve ici le 2ème AOP utilisé pour l'amplification
En 3 on a la sortie du montage qui se dirige vers le point 4 du schéma suivant.
On voit le cavalier JP3 qui permet de modifier le gain.



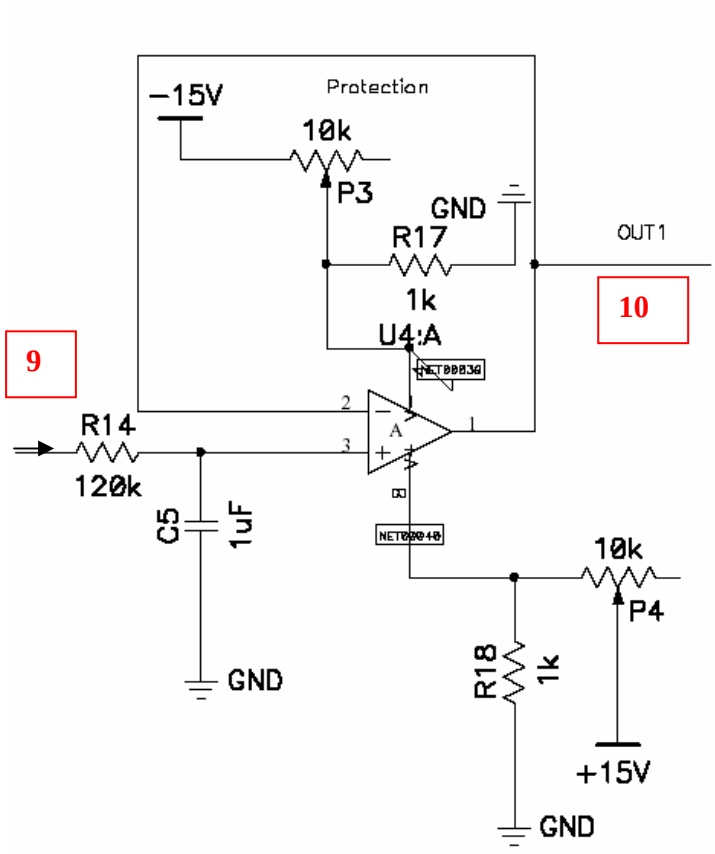
On trouve ici le filtre à 40kHz.
En 5 on a la sortie du montage qui va au point 6 du schéma suivant

Réglage de la bande passante

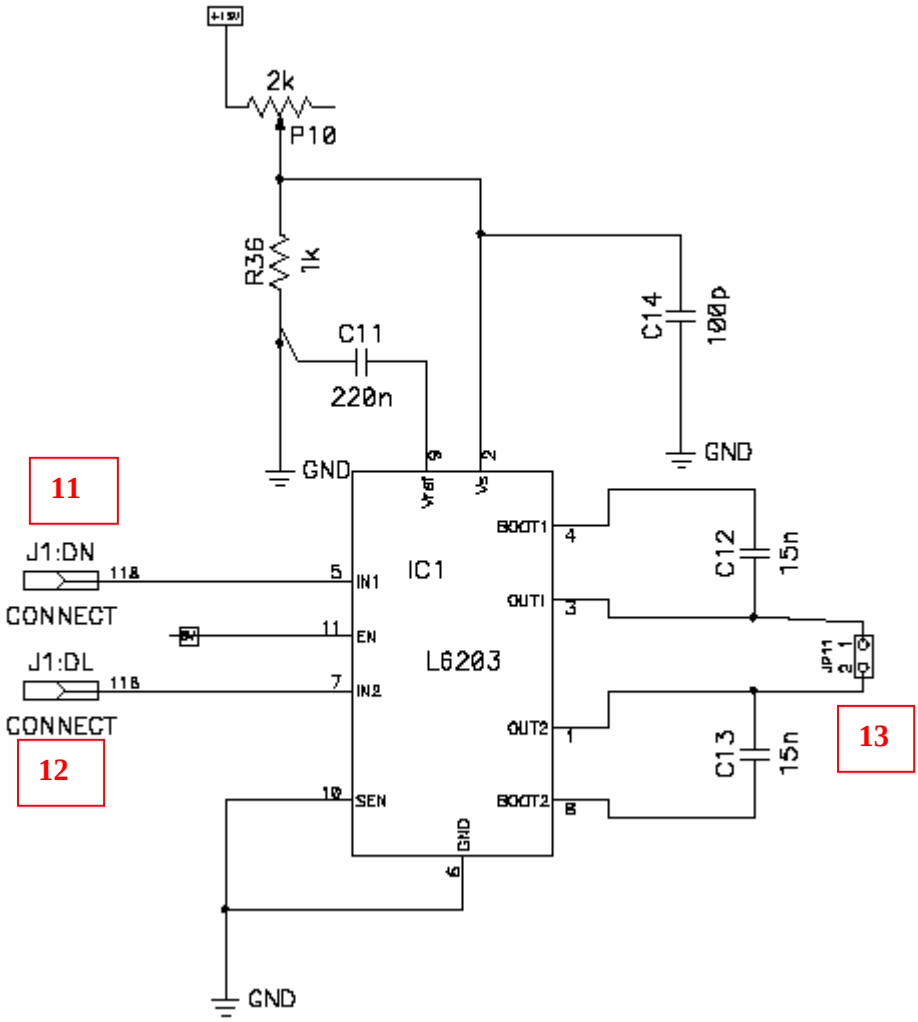
Réglage de la fréquence centrale



Sur le circuit précédent: on trouve le circuit de redressement double alternance sans seuil et le 2ème AOP pour amplifier le signal.
En 8 le signal se dirige sur point 9 du schéma suivant.
En 7 on a un cavalier qui permet de faire varier le gain de l'amplification. On peut donc avoir soit un gain de 60 ou de 27. (ce cavalier n'était pas présent sur la carte prototype)



Ici on a le filtre pour obtenir la tension continue, et le circuit de protection pour limiter la tension entre 0 et +5V.
Les 2 potentiomètres permettent un réglage fin de la tension de sortie de l'AOP.
En 10 on a la sortie du signal que l'on peut faire aller à l'entrée analogique du microprocesseur.

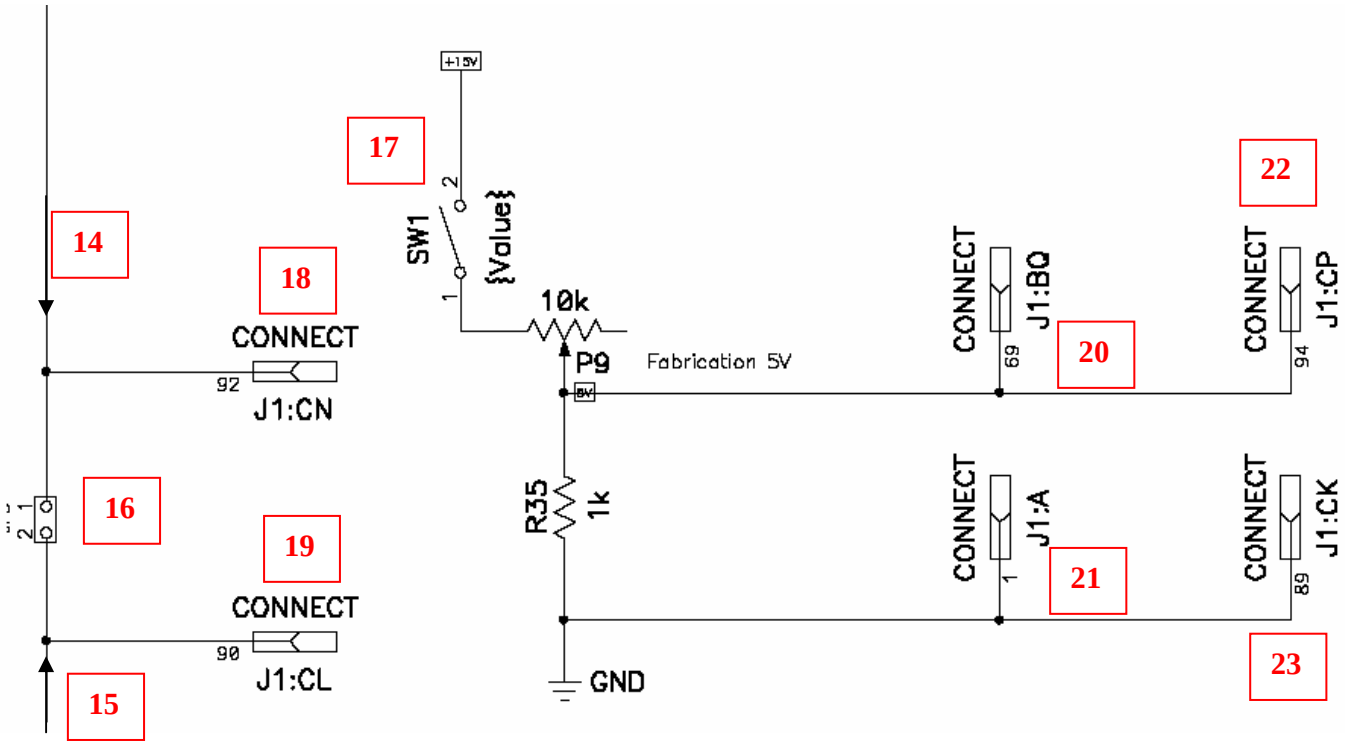


On trouve ici le circuit du driver L6203

En 11 on trouve le connecteur provenant du signal PWM1
Patte 118 du microcontrôleur

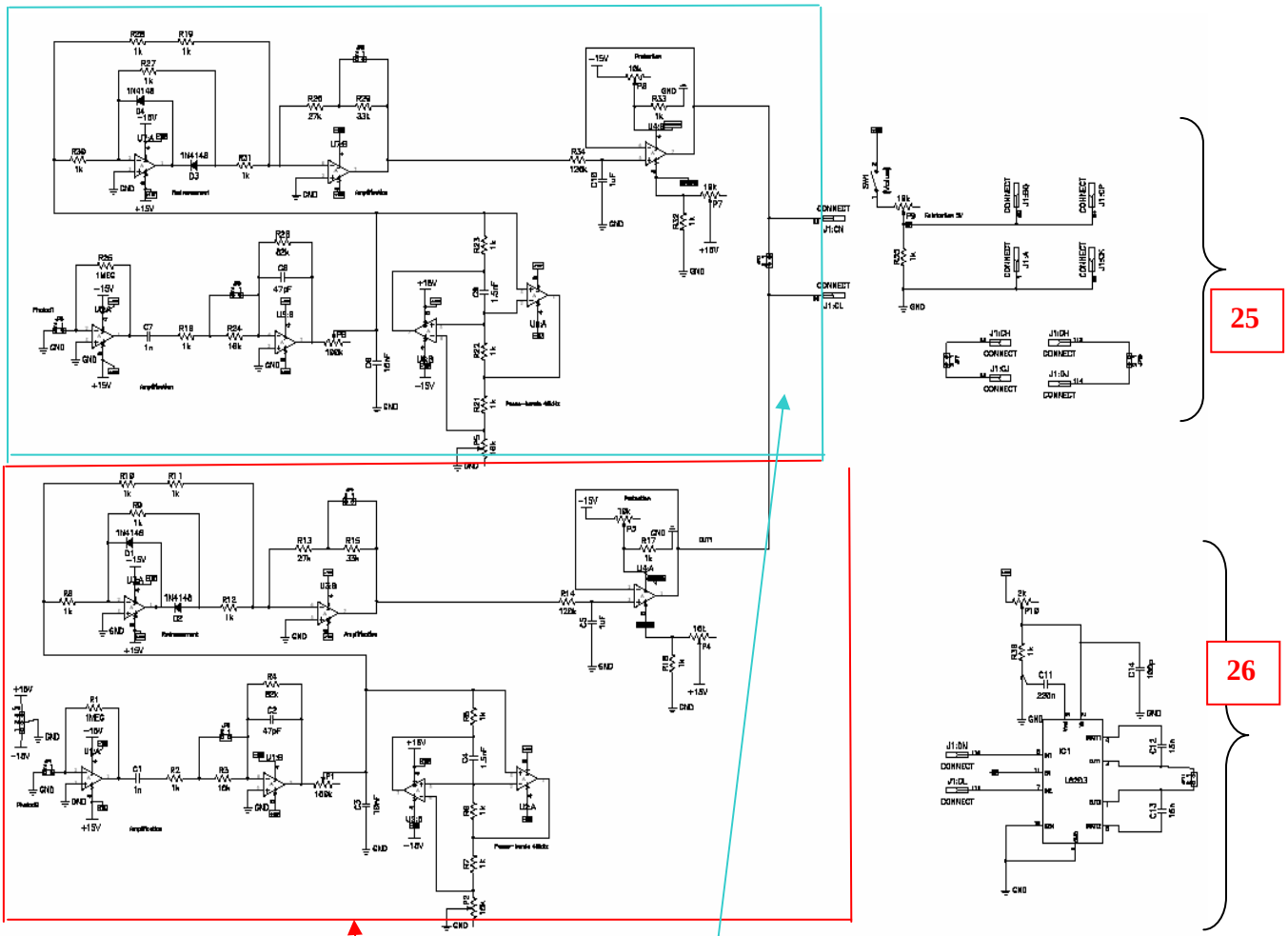
En 12 c'est l'arrivée du 2ème signal PWM.
Patte 116 du microcontrôleur

En 13 c'est le connecteur pour le moteur



En 14 on a l'arrivée du signal issu du limiteur de tension de la photodiode 1, et en 15 c'est celui de la photodiode 3.
En 16 on a un connecteur sur lequel on pourra brancher les entrées d'oscilloscope pour vérifier le bon fonctionnement de la partie traitant le signal issu des photodiodes.
En 17 on a l'interrupteur général du montage.
En 18 on a la patte 92 qui correspond à l'entrée P5.0 du microcontrôleur (entrée analogique).
En 19 on a la patte 90 qui correspond à l'entrée P5.1 du microcontrôleur (entrée analogique).
En 20 on a la patte 69 qui correspond à l'alimentation VCC du microcontrôleur.
En 21 on a la patte 1 qui correspond à la masse du microcontrôleur (on doit y mettre 5V).
En 22 on a la patte 94 qui correspond à VAREF du microcontrôleur (on doit y mettre 5V).
En 23 on a la patte 89 qui correspond à VAGND du microcontrôleur (on doit y mettre 0V).

Finalement on obtient ce schéma:

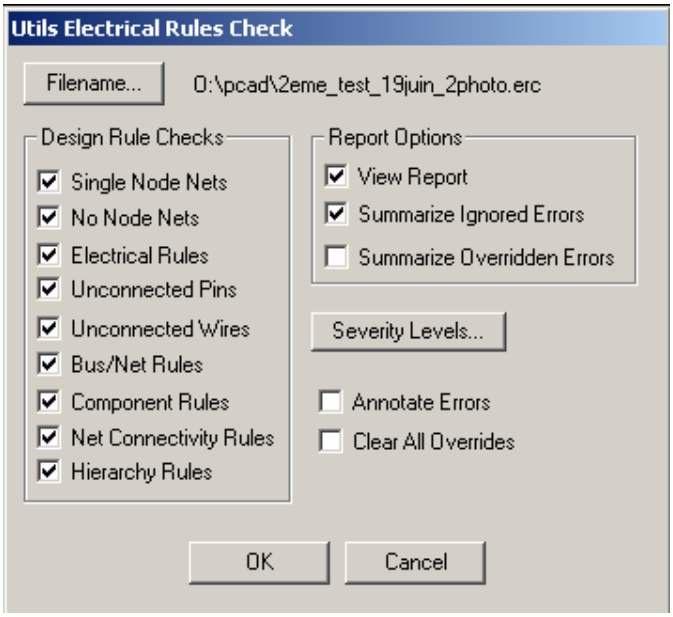


Traitement du signal de la photodiode 2

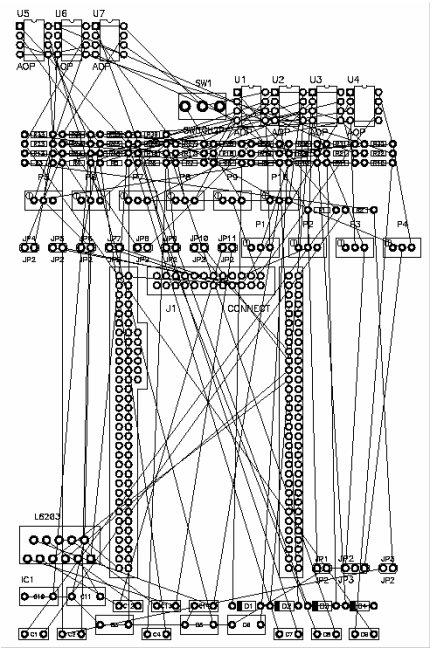
Traitement du signal de la photodiode 1

En 25 on a les connexions sur le microcontrôleur
En 26 il s'agit du driver pour le moteur.

Des options sont à définir pour augmenter la taille des pistes d'alimentation.
Il faut ensuite vérifier que le circuit ne comporte pas d'erreur: dans Utils → ERC…
La fenêtre suivante apparaît:



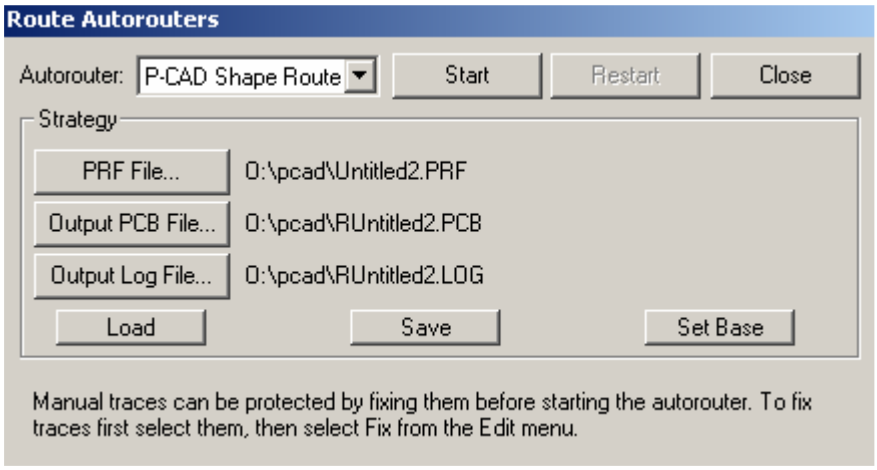
Valider le choix et consulter le fichier texte pour voir les erreurs à corriger.
Dès lors qu'il n'y a plus d'erreur on peut générer le fichier netlist: dans le menu: *Utils* → *Generate netlist…*
Puis pour le routage aller dans *PCAD PCB.*
Aller au menu: *Library Setup* : et ajouter les différentes librairies utiles à ce circuit
Puis aller au menu: *Utils* → *Load Netlist* + choisir le fichier **2eme_test_19juin_2photo.net**



Il faut maintenant placer les composants (on pourrait mettre des composants sur une face: comme le microcontrôleur et les autres composants sur une autre couche).

7.3.9.3. Le routage

Pour lancer le routage aller dans le menu *Route* → *autoroute*…Et la fenêtre suivante apparaît:



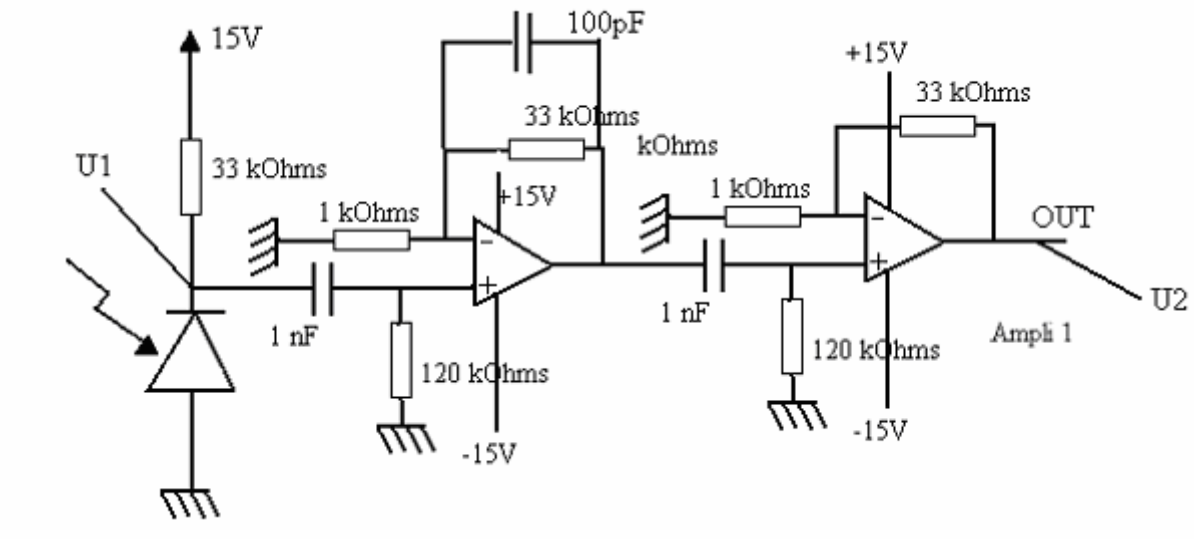
Après validation et choix du fichier de destination le routage se réalise.

La réalisation sur PCAD est maintenant terminée. Pour réaliser la plaque sur circuit imprimé il suffit de donner certains fichiers à M. Kiesling, la conception s'est arrêtée là car il était trop tard pour souder les composants et de plus il aurait fallu tenir compte de la place dont on a à disposition sur le "bras".
(Une vérification complète devrait être faite avec M Kiesling).

7.3.10. Amélioration

- **Amélioration de la pré amplification :**

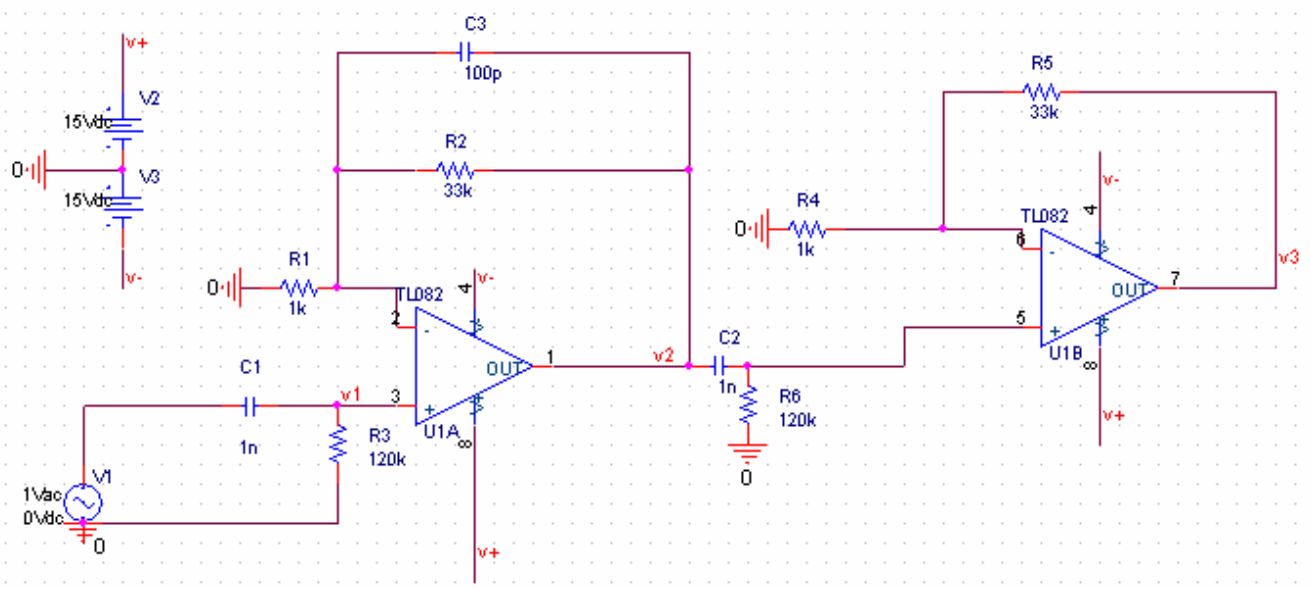
Le montage présenté en partie pré amplification pose certains problèmes : il n'est pas assez rapide. Une capacité parasite augmente avec la fréquence du signal provenant de la photodiode. Le domaine d'utilisation de ce circuit est en dessous de la centaine de kHz environ.
Un montage qui améliore ce défaut est présenté en dessous. La photodiode est polarisée. Les AOP sont de type TL082 (pour des raisons qui seront vues plus loin il serait préférable d'utiliser des AOP LM6142)



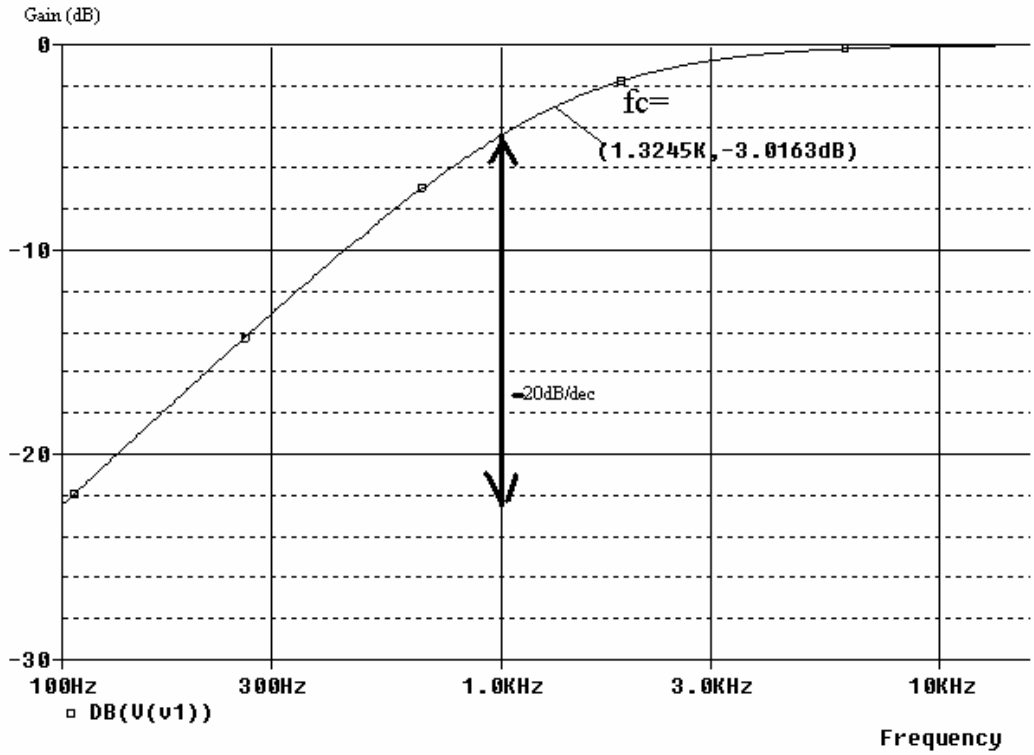
Sur le schéma on remarque que nous avons deux filtres RC (120kOhms, 1nF) qui ont une fréquence de coupure de 1/(2*pi*R*C)=1326 Hz, ce qui nous permet de diminuer l'influence de la fréquence du réseau (50Hz). De plus nous utilisons deux AOP avec un produit gain*bande passante de 17MHz ce qui nous fait une fréquence de coupure haute de 3 000 000/34 =88kHz. (Gain=1+R2/R1 = 1+ 33k/1k=34)
Ces deux fréquences de coupure doivent nous donner une allure de passe bande lors de la réponse en amplitude.

C'est ce qui va être vérifié en faisant le diagramme de Bode de l'installation :
L'amplification est de Gain(1er AOP) * Gain(2ème AOP) = 34*34= 1156

Simulation sous Pspice (version 9.23) et détermination de la réponse en fréquence en v1, v2 et v3 sur le schéma suivant.

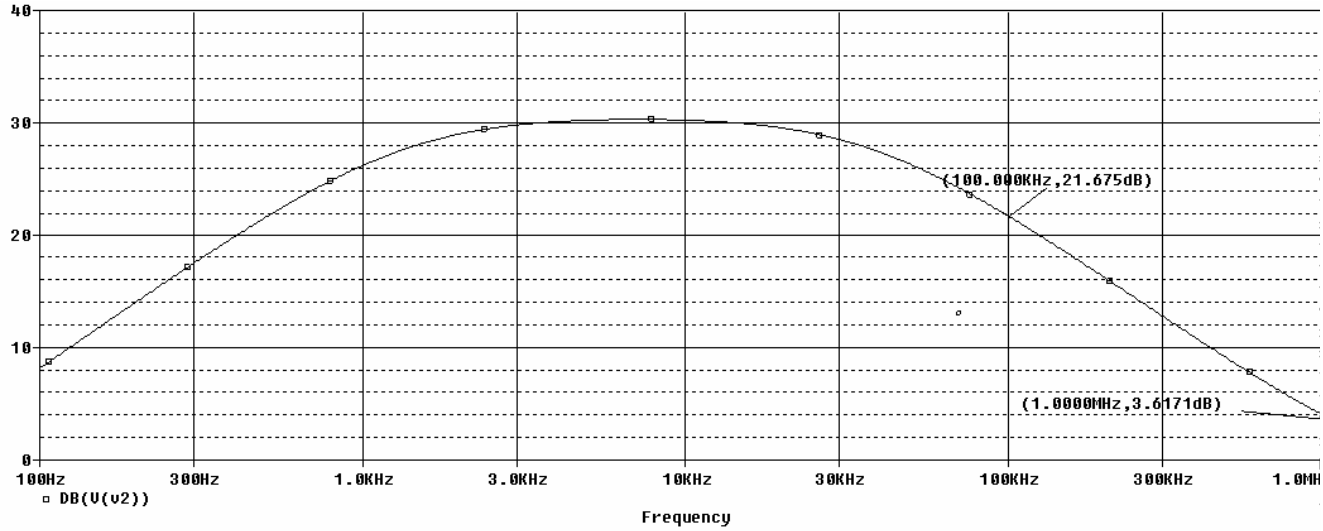


Réponse en fréquence du signal v1 : (filtre passe-haut RC, fc=1/(2*PI*120k*1nF)=1326Hz)



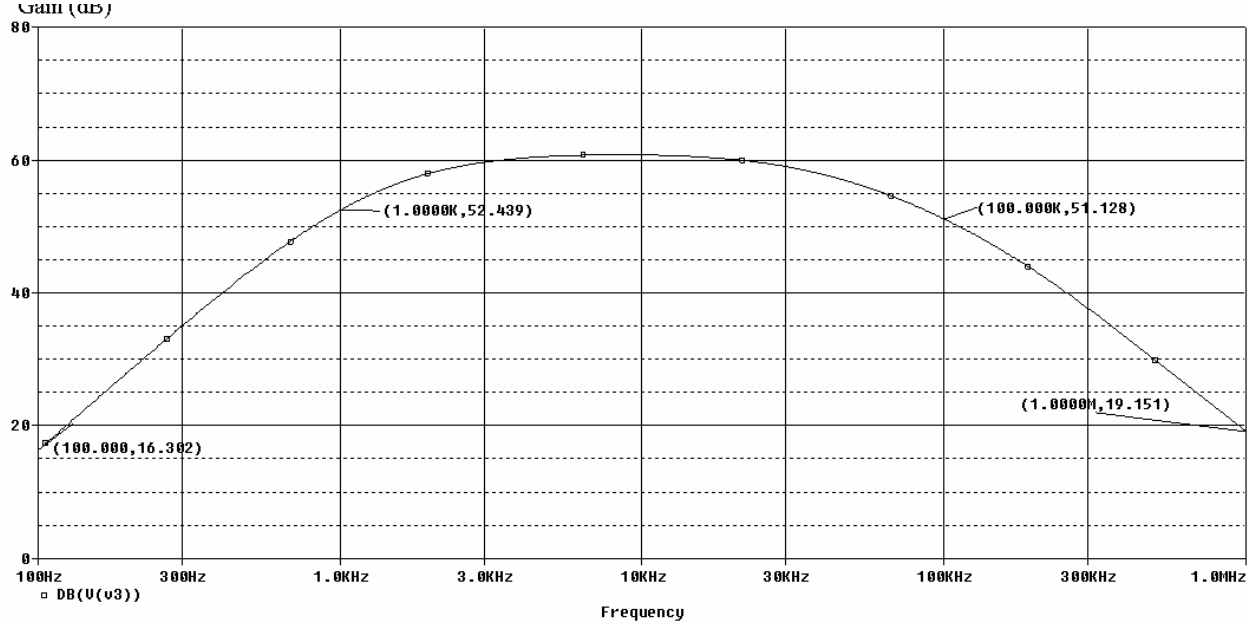
Cutoff_Highpass_3dB(V(v1)) 1.32947k

Gain en fonction de la fréquence du signal v2 : (passe-bande avec une bande passante de 39.3kHz)



Bandwidth_Bandpass_3dB(V(v2))	39.35949k
Cutoff_Highpass_3dB(V(v2))	1.24533k
Cutoff_Lowpass_3dB(V(v2))	40.60483k

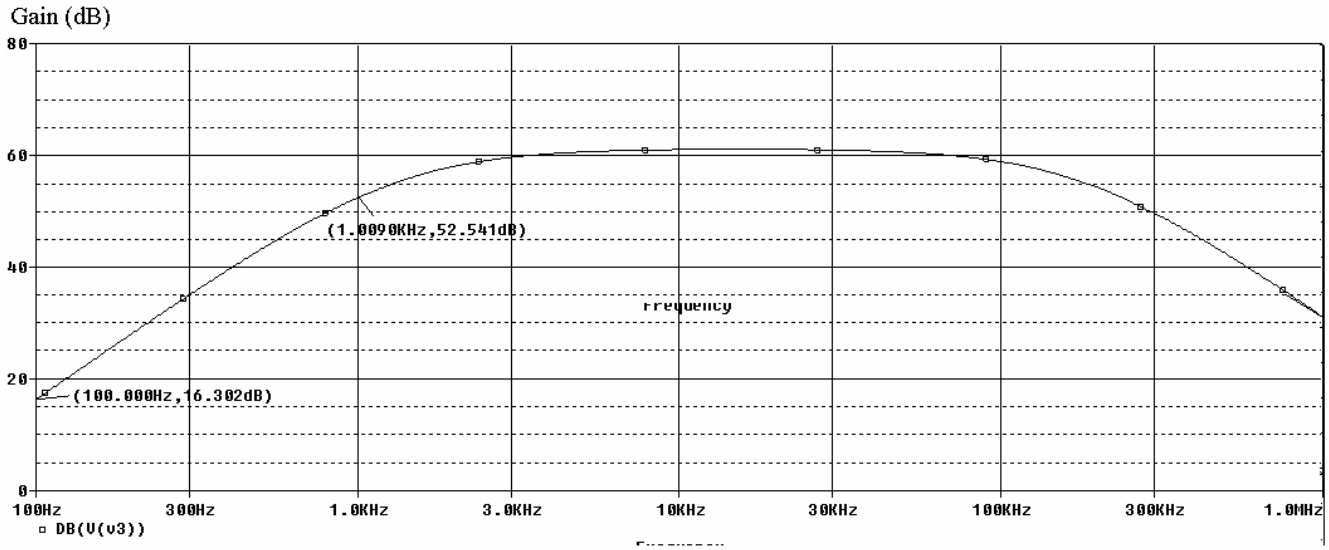
Les pentes sont de –20dB par décade.
Réponse en fréquence du signal v3 : (passe-bande avec une bande passante de 39.8kHz)



Les pentes sont de –36dB et –32dB par décade.
La 1ère pente est maintenant de –36dB par décade : cela est dû au fait qu'on ajoute un filtre passe-haut avec la même fréquence de coupure. Et la 2ème pente passe à –32dB. (Cela est dû à l'AOP et le filtre parasite du à la capacité de 100pF).

Bandwidth_Bandpass_3dB(V(v3))	37.97179k
Cutoff_Highpass_3dB(V(v3))	1.91007k
Cutoff_Lowpass_3dB(V(v3))	39.88185k

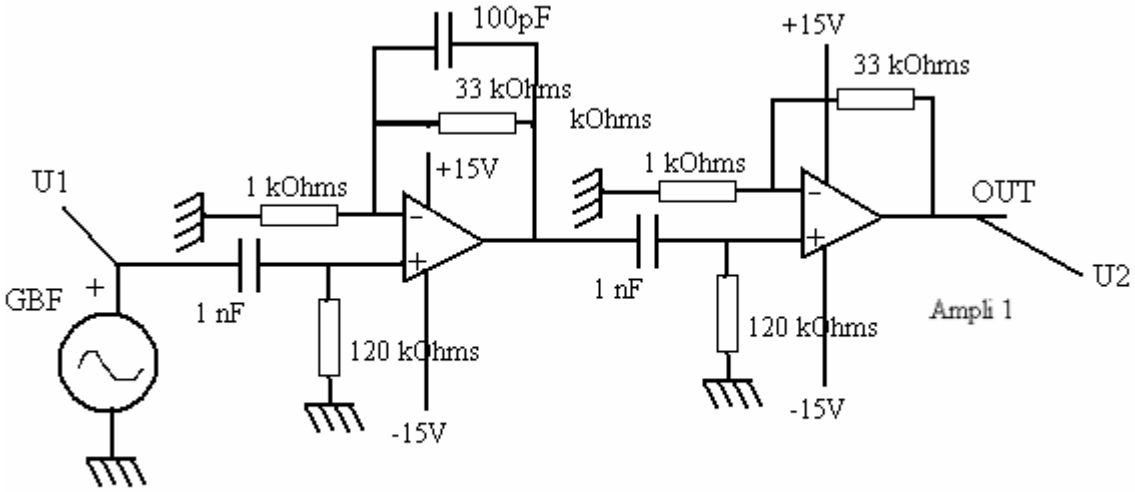
Détermination de l'effet de la capacité de 100pF.
Effet de la capacité de 100pF sur le 1er AOP : (on la retire)



Bandwidth_Bandpass_3dB(V(v3))	116.05740k
Cutoff_Highpass_3dB(V(v3))	2.01485k
Cutoff_Lowpass_3dB(V(v3))	118.07225k

La capacité de 100pF diminue la bande-passante (cela ajoute –20dB/dec car c'est un passe-bas) car sa fréquence de coupure est vers 48kHz.

Schéma pour la mesure du diagramme de Bode :



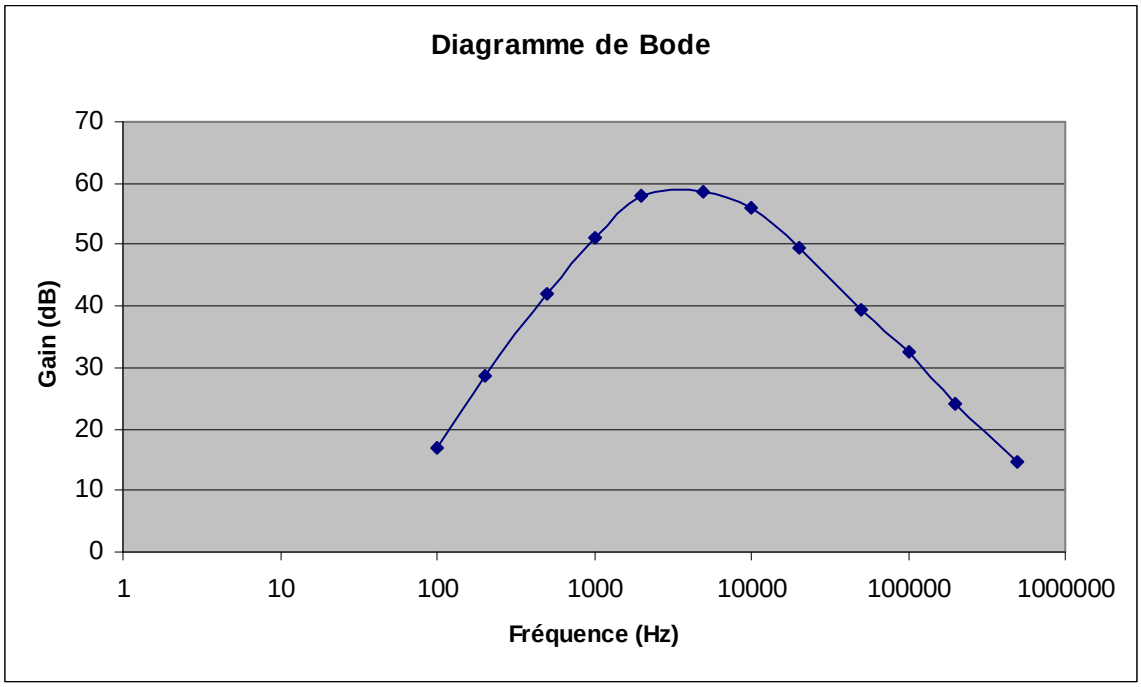
Rq: Les amplis Op sont des TL082

On place un GBF en entrée du montage. (On atténue la tension afin de ne pas saturer à la sortie des amplis). (Pour ce test les AOP TL082 ont été utilisés).

Pour mesurer la réponse en amplitude : on mesure la tension U1 et U2 selon la fréquence du GBF. (Rq : à la place de la photodiode on met un générateur de basse fréquence GBF pour être plus précis dans la mesure de cette réponse d'amplitude en fonction de la fréquence).

Tableau de mesure :

Fréquence (Hz)	100	200	500	1000	2000	5000	10000	20000	50000	100000	200000	500000
U1 (V)	0.04	0.03	0.03	0.04	0.03	0.03	0.03	0.04	0.03	0.04	0.04	0.03
U2 (V)	0.28	0.8	3.8	14.4	24	26	15	10	2.8	1.7	0.65	0.16
Gain (dB)	16.902	28.519	42.053	51.126	58.062	58.757	53.979	47.959	39.401	32.568	24.217	14.54

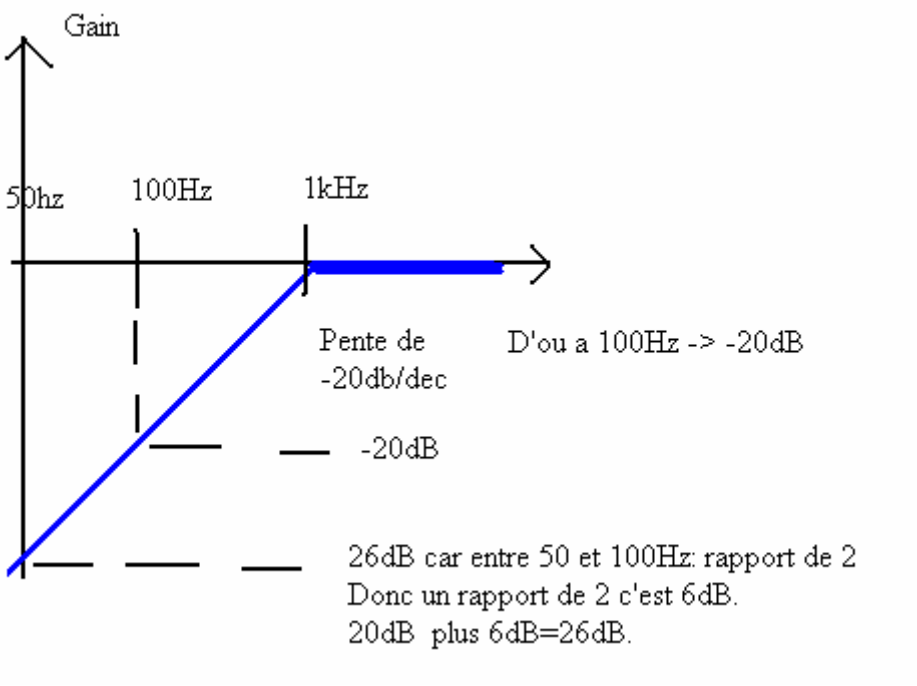


Commentaire :

On trouve bien une allure de la réponse de type passe-bande.
La 1ère fréquence de coupure provient des filtres C-R (1.3kHz), la 2ème fréquence de coupure provient de la limitation des AOP dû à leur produit gain * bande passante de 3Mhz.et de la capacité de 100pF, elle se situe vers 20kHz.

Effet de la capacité de 100pF : cela fait un filtre passe-bas de 48.2kHz. Le but de cette capacité est d'empêcher l'AOP d'entrer en oscillation.
(hors on cherche à avoir une fréquence de coupure haute de l'ordre de 40kHz, cela a pour effet de faire « gagner » 20dB d'atténuation par décade).
(fc=1/(2*PI*33k*100p= 48.2kHz).

Sur le schéma suivant on voit l'effet d'un filtre passe-haut R-C avec fc=1kHz qui atténue de 26dB le 50Hz. (Il faudrait transposer ce résultat pour un filtre à 1.3kHz)



Détermination du condensateur de 1nF :
fc=1/(2*Pi*RCin)
D'où : Cin=1/(2*PI*fcoupure*R)=1nF
Avec R=120k

Cin est un condensateur de découplage, avec R=120k cela forme notre passe-haut à 1.3Khz.

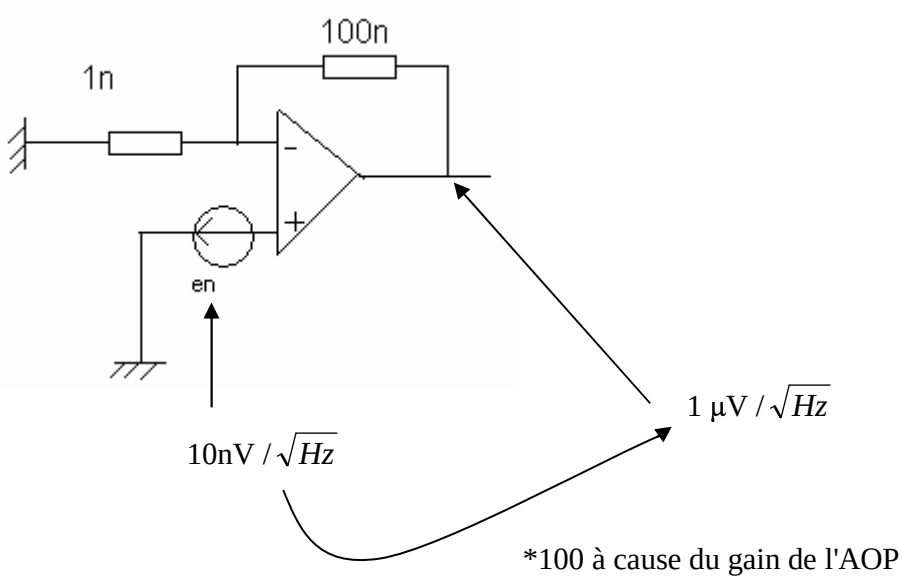
Autres améliorations à porter sur le circuit :
- Le filtre à 40kHz n'est pas assez performant : s'il venait à y avoir une source infrarouge proche des 40kHz avec une puissance plus grande que la vraie source alors le capteur se dirigerait sur la source la plus importante en terme de puissance.
Il aurait été intéressant de réaliser un filtre numérique avec un processeur DSP programmé. Mais cela n'a pas été vu en cours.

- Le filtre permettant d'obtenir la tension continue n'est pas assez rapide. Il faudrait envisager d'utiliser le montage à AOP avec en contre réaction une résistance et une capacité. Cela ferait un filtre passe-bas

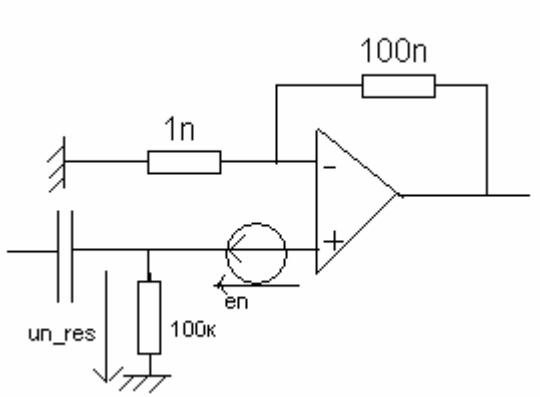
- Je ne connaissais pas les méthodes généralement employées pour limiter la tension entre 0 et 5V avant l'entrée d'un microcontrôleur, une autre méthode serait d'utiliser des diodes Zéner (leur tension de jonction devrait être choisie basse).

Amélioration vis-à-vis du choix des composants :
Les composants génèrent du bruit sur les signaux ce qui explique qu'on ne puisse pas faire de gain démesuré.

Provenance du bruit dans les signaux:
Les AOP génèrent du bruit.
Les AOP ont un certain facteur de bruit noté e_n dans les datasheets.
Par exemple: un AOP avec un e_n de 10nV/$\sqrt{Hz}$. Ce bruit dépend du gain de l'ampli.



Les résistances génèrent aussi du bruit:



Quelques valeurs de bruit générés par des résistances:

$4nV/\sqrt{Hz}$ → R=1kΩ

$40nV/\sqrt{Hz}$ → R=100kΩ

De plus quand on ajoute le bruit issu des résistances et des AOP ce n'est pas leur tension qu'on additionne mais leur puissance. (La puissance est proportionnelle à la bande passante).
Donc pour diminuer ce bruit il faut choisir des AOP avec un e_n le plus faible possible.

7.3.11. Problèmes rencontrés

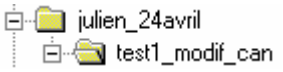
La partie pré amplification permet de recevoir des signaux allant jusqu'à 80-100m. Mais dès lors que l'on connecte les circuits de filtrage, de redressement, de filtrage pour obtenir la tension continue et de limitation de tension alors le signal est fortement atténué. C'est pourquoi des modifications sont certainement à effectuer.

8. Le programme

Cette partie traite des programmes qui ont été utilisés pour réaliser ce projet. Le 1er programme portera sur l'acquisition d'un signal. Le second traite de l'utilisation d'une unité PWM. Ces 2 programmes permettent de réaliser le programme final dédié au contrôle du système de tracking optique.

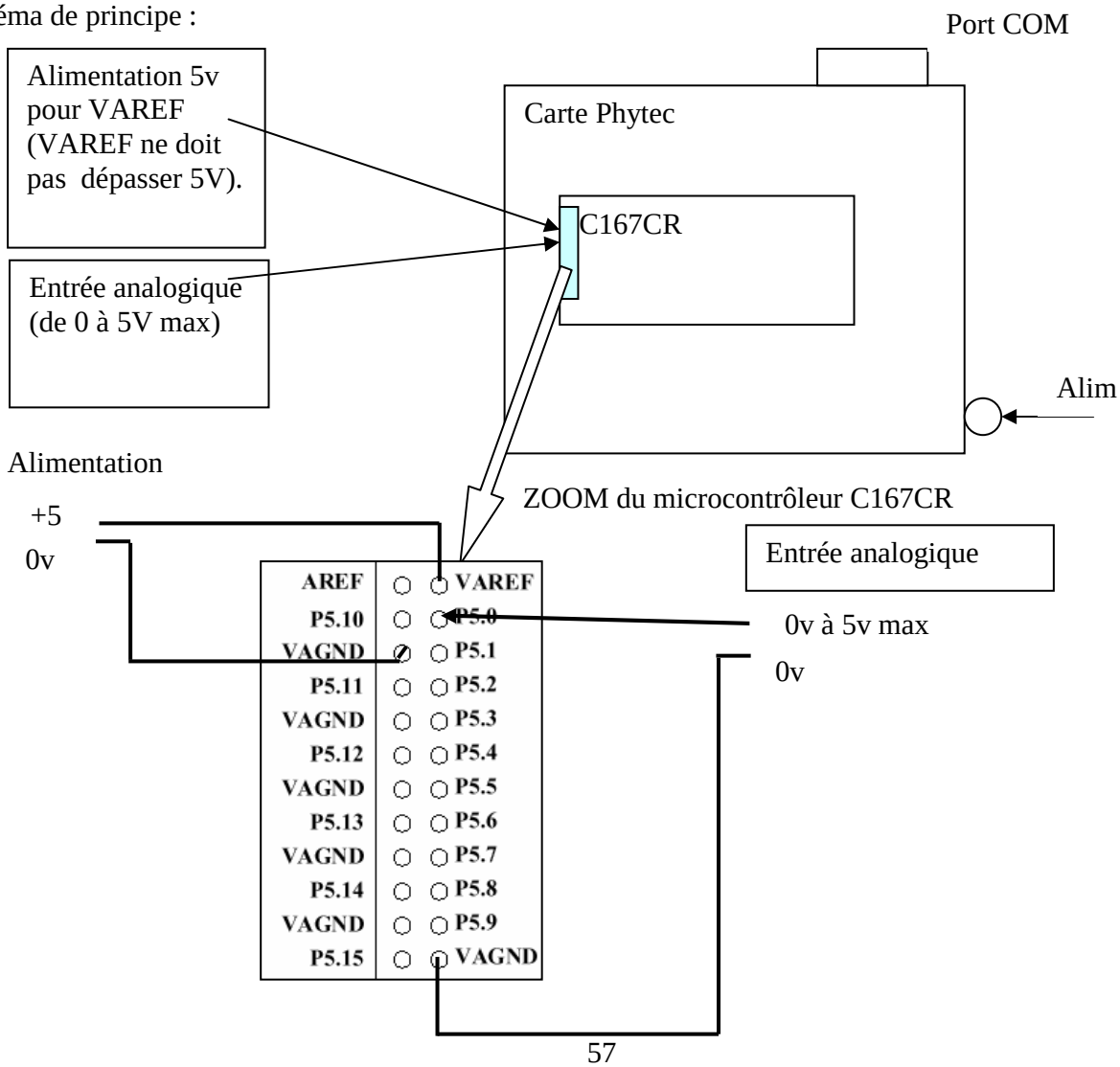
8.1. Acquisition d'un signal

Nom du projet :



But :
Ce programme issu des exemples fournit par Phytec permet de tester la conversion analogique numérique.
Le canal 0 (P5.0) est utilisé comme entrée analogique.
VAREF doit être à 5V, VAREF sert de référence.
Utilité de cet exemple : cela permettra d'acquérir la tension issue des photodiodes après le traitement de leurs signaux.

Schéma de principe :



Comme on le voit dans cet exemple P5.0 est relié à une source de tension sur lequel on fait varier la tension de 0 à 5V max.
Le programme pour réaliser cette acquisition de tension est le suivant:
Programme source : ADC1.C

```
/*****************************************************************************
 ADC1.c
 A/D-Converter Routine for own Channel
*****************************************************************************/

#include <REG167.H>                 /* Register Set of 80C167 controller   */
#include <stdio.h>                    /* Standard I/O functions              */

unsigned int Result;


/***** main program ***********************************************************/

void main (void)  {
  while(1)
  {                                                  /* init AD-Converter  */
    ADCON = 0x00;                 /* A/D-Converter: Channel0, Single     */
                                  /*  Choix de P5.0                        */
                                  /* Channel Conversion                */
    ADST = 1;                     /* Debut conversion                     */

      while(ADBSY);               /* Attente fin de conversion            */

    Result = ADDAT & 0x03FF;      /*Resultat de la conversion       *
  }
}
```

Test :
Soit VAREF fixé à 4.99V, et la tension sur l'entrée analogique P5.0 est fixée à 4.00V.
Testons le programme ci-dessus dans CrossView. On peut en effet voir le résultat qui se trouve dans les différents registres. C'est dans ADDAT qu'est mis le résultat.
Le registre ADDAT nous donne = (0334)hexa soit (820)décimal.

Quelle est la résolution du convertisseur ?
Le résultat de la conversion se trouve dans le registre ADDAT et 10 bits lui sont réservés pour le résultat. (bits 0 à 9). Les autres bits (10 à 15) ne nous intéressent pas pour l'instant (c'est pourquoi on fait une masquage dans le programme).
Avec 10 bits on dispose d'une valeur qui va varier de 0 à 1023.
Pour 4.99V on a 1024 bits: donc pour 1bit on a : 4.99/1024=4.873mV.
La précision du résultat dans ADDAT est donc de 4.873mV par bit.
(Rq : Avec un convertisseur de 12 bits on aurait eu 4.99/4096= 1.218mV)

Vérification de la tension lue sur le canal P5.0 :
Pour 4.99v -> 1024bit
ADDAT nous donne 820 dans notre cas.
Donc U=820*4.99/1024=3.995v

Comparaison :
Sur le canal 0 une tension de 4v a été fixée, et après la conversion en numérique on obtient 3.995v. L'erreur dans cet exemple est minime.
La précision de 4.873mV posera problème si le signal analogique est faible (si l'émetteur infrarouge est placé trop loin du récepteur).

Le test a été effectué avec le signal traité de la photodiode.

8.2. Utilisation d'une unité PWM

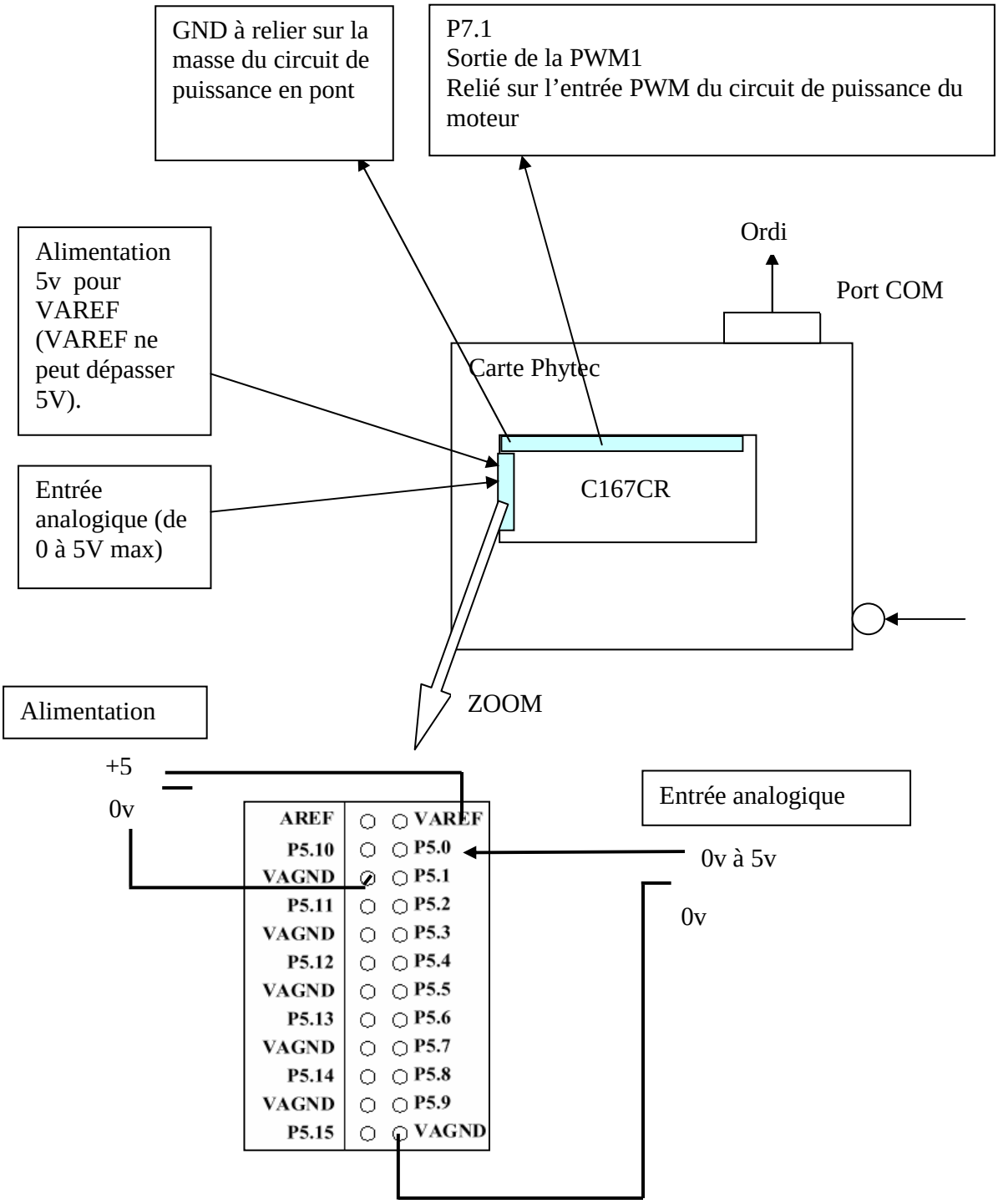
Le but de cette partie est de commander le moteur à courant continu grâce à une unité PWM du microcontrôleur. Le montage de puissance pour le moteur azimut a donc été réalisé (avec le driver). Le driver est commandé par 1 unité PWM venant du port P7.1. Le rapport cyclique de la PWM varie selon la tension qui est appliquée sur l'entrée analogique P5.0.

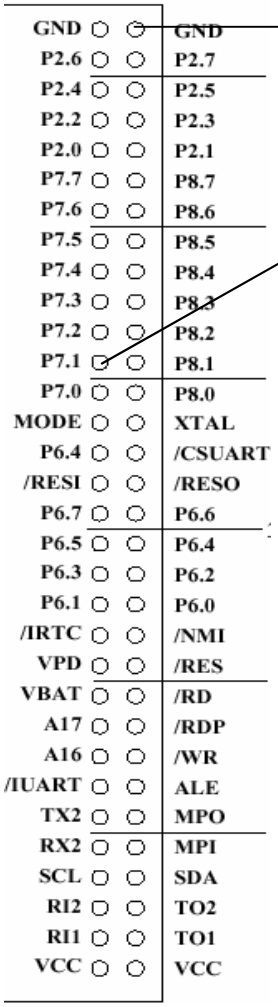
Nom du projet :

TASKING EDE [C166/ST10 - C:\c166\examples\julien_7mai_2pwm_2can_essai_regul_moteur\test1_2pwm_2can\test1.pjt] - [C:\c166\examples\julien_7mai_2pwm...

Le programme suivant produit une PWM qui varie en fonction de la tension qu'on entre sur le CAN. Le montage de puissance est alimenté en 12V.

Schéma du montage :





GND vers masse du circuit de puissance en pont

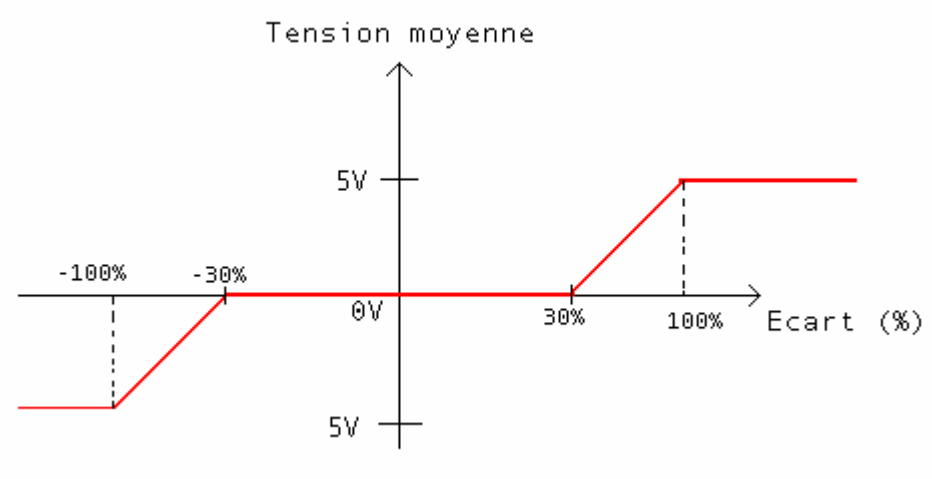
P7.1 vers entrée PWM1 du circuit de puissance en pont

Programme :

```
/*********************************************************************
MARGUET Julien
Ce programme fait varier le rapport cyclique de la PWM1 de P7.1 selon la tension qu'on applique sur l'entrée analogique de P5.0

La PWM varie ainsi de : 0 à 5V de moyenne

U
|     __________    __________
|     |         |   |         |
| __  |         |__ |         |__
-------------------------------------------------> t
 La fréquence est fixée à 250Hz car:
 Frequence = 20 MHz / (2 * (40000-1)) = 250 Hz
 *********************************************************************/

#include <REG167.H>         /* Register define for C167             */
_sfrbit DP71 _atbit(DP7,1);

/* --------------------------AD-Converter---------------------------------------------------------------------------- */
/*La fonction ad_conv convertit la tension lue sur P5.0                                                              */
/*Cette fonction retourne la nouvelle valeur de PW1 selon la tension qui a été lue                 */
/* Si la tension lue est de 0V alors il faut que la PWM s'arrête. Pour cela il faut que PW1=40000        */
/* Si la tension lue est de 5V il faut que la PWM soit très rapide : soit PW1=0                          */
/*Si l'on met PW1=0 alors la valeur moyenne de la tension sera max donc de 5V                 */
/*Si PW1=40000 alors la tension moyenne sera de 0V                                                     */
unsigned int ad_conv (unsigned char ADC_Channel)
{
unsigned int Result;
unsigned long Result_PW1;                   //type long  car on fait 40000*valeur...le type int est limité à 65536U

/* init AD-Converter                                                                                    */
 ADCON = 0x00 + ADC_Channel;          /* A/D-Converter: Canal 0          */
 ADST = 1;                                    /*Debut  Conversion               */
while(ADBSY);
Result = ADDAT & 0x03FF;                    /*Result ne varie que entre 0 et 1023 max*/
Result_PW1=40000-((Result*40000)/1023);   //Result_PW1 varie entre 0 et 40000
                    /*donc la tension moyenne sur P7.1 varie entre 0 et 5V      */


 return Result_PW1;
}

/* ------ main Programme: ----------------------------------------  */
void main (void)
{
   DP71  = 1;                /*DP71 : definit en tant que sortie PWM              */
   PM1   = 1;                /* permet le choix du mode : Center Aligned          */
   PP1   = 40000;            /* Load period register                              */
   PW1   = 20000;          /* Load pulse width register                           */
   PEN1  = 1;               /* P7.1 autorise le signal de sortie                  */
   PTR1  = 1;               /* Lance le PWM-Timer 1                               */

   while (1)
          {
                    PW1 = (ad_conv(0));                 //retourne nouvelle valeur de PW1
          }
}
/* -- End of program ---------------------------------------------- */
```

8.3. Le programme final

Le programme qui va suivre est celui qui a été utilisé pour réguler la position du capteur optique en face de la source infrarouge.
Les étapes de ce programme sont les suivantes :
Il faut d'abord lire les tensions provenant des 2 photodiodes sur P5.0 et P5.1.
Selon les valeurs de ces 2 photodiodes il faut positionner le moteur, plusieurs tests sont alors nécessaires.
Le 1er test à effectuer est de vérifier si le capteur optique détecte un signal. Si ce n'est pas le cas on fait tourner le moteur jusqu'à ce qu'on détecte un signal.
De plus dans ce test on peut régler le niveau de détection selon le bruit provenant du signal de la photodiode. Si le circuit fournit des signaux ayant beaucoup de bruit on peut mesurer ce bruit à l'oscilloscope et en déduire que si le signal reçu n'est pas au delà de cette valeur de bruit il ne faut pas en prendre compte.
Quand un signal commence à être perçu il faut vérifier laquelle des 2 photodiodes fournit le plus grand signal. Celle qui fournira le plus grand signal déterminera le sens du moteur.
Quand les valeurs des signaux de ces 2 photodiodes seront à peu près du même ordre de grandeur c'est que notre capteur optique aura été positionné en face de la source infrarouge.
Le moteur tourne assez lentement pour avoir le temps de détecter un signal. Une régulation avec un correcteur proportionnel a été appliqué :
Le moteur tourne à vitesse constante s'il ne détecte rien. Plus le signal devient fort sur les photodiodes et plus cela veut dire qu'on se rapproche de la source, ainsi plus la vitesse du moteur doit être réduite.
Dans le programme est calculé l'écart en pourcentage entre les 2 valeurs de tension fournies par les 2 photodiodes. (variable : ecart).
Ecart=(TensionPhotodiode1-TensionPhotodiode2)*100/1023.
L'écart varie donc entre –100% et +100%.
Si l'écart est compris entre +-30% alors le moteur est arrêté car l'alignement du capteur et de la source est bon.
Dans le graphique ci-dessous on peut voir la régulation de la vitesse en fonction de l'écart. (Le graph n'est pas à l'échelle).
Si l'écart est positif le moteur devra tourner dans le sens des aiguilles d'une montre. S'il est négatif ce sera dans le sens inverse. Si la valeur absolue de l'écart est grande alors la tension moyenne à appliquer devra être grande.



Cependant mettre 5V de tension moyenne ferait aller le moteur trop vite.

Plutôt que de parler en terme de tension la suite du développement parlera en terme de valeur à entrer dans PW1 ou PW2 pour faire varier le rapport cyclique.
PW1 et PW2 peuvent varier entre 0 et 40000

Si PW1=		Vitesse :
0	→	Max
40000	→	Arrêt du moteur

Calcul de la pente du graph précédent:
Elle est de la forme:

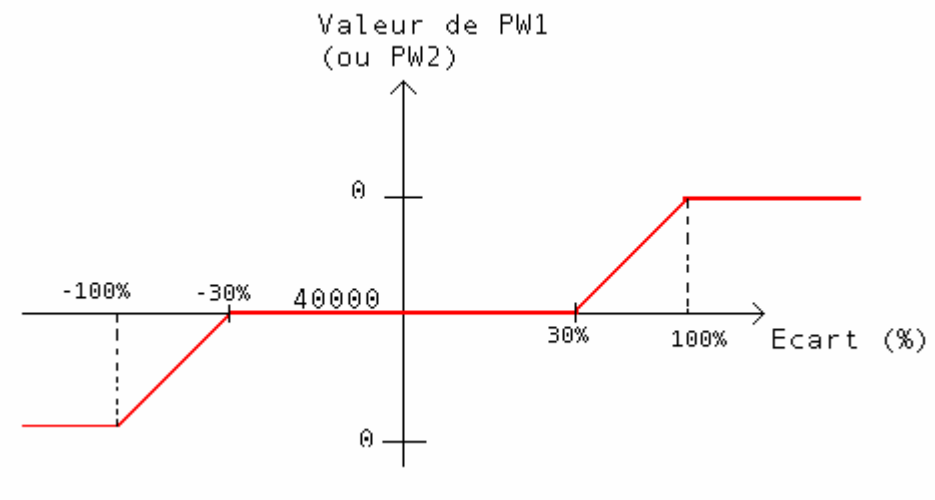
Y=AX+B
A=(40000-0)/(100%-30%)
B=17100
Y=A*ecart-17100=571*ecart-17100

Calcul de PW1 (ou PW2) selon l'écart:

40000-(ecart*571-17100)

Exemple:

Si l'écart est de 50% alors la valeur de PW1 sera:
40000-(50*571-17100)=28450
Si l'écart est de 30%: PW1=39970
Si l'écart est de 100%: PW1=0



Code source:

```
/* ********************************************************************
MARGUET Julien
Le 7 mai 2003
Programme pour réguler la position du moteur azimut selon les signaux fournits par les 2 photodiodes en P5.0 et P5.1
Test si le niveau de tension fournit par ces 2 photodiodes est assez élevé.
Utilise 2 CAN et 2 unités PWM
Une PWM fait tounrer le moteur dans un sens, et l'autre PWM fait tourner le moteur dans l'autre sens.
La fréquence de PWM est fixée à 250Hz car:
 f = 20 MHz / (2 * (40000-1)) = 250 Hz

******************************************************************** */

/* -------Including Files: ---------------------------------------- */
#include <REG167.H>       /* Register defin for C167              */
#include <stdlib.h>                  /*pour valeur absolue*/

_sfrbit DP71 _atbit(DP7,1);          /* Bitname DP71 for DP7.1 define          */
_sfrbit DP72 _atbit(DP7,2);          /* _atbit: indique que c'est le bit 2 de DP7          */
                                     /* DP7 est un registre SFR: on a accés au bit 0 à 7   */
                                     /*       _sfrbit DP72: definit un bit nommé DP72      */

/* -------------------------AD-Converter------------------------------------------------ */
/*Conversion de la tension sur le canal  0 ou 1  (P5.0 ou P5.1)                          */
unsigned int ad_conv (unsigned char ADC_Channel)
{
unsigned int Result;
/* init AD-Converter                                                                     */
 ADCON = 0x00 + ADC_Channel; /* A/D-Converter: Channel 0, Single                         */
ADST = 1;                                    /* Conversion start                      */
 while(ADBSY);
Result = ADDAT & 0x03FF;          /*Result ne varie que entre 0 et 1023 */
 return Result;
}
/* -------------------------FIN AD-Converter--------------------------          */


/* -------------------------CALCUL LA FORCE de PWM            ----------------------------      */
/*Selon l'ecart de tensions entre les 2 photodiodes il faut calculer la valeur à rechager dans PW1 ou PW2 */
/*pour définir le nouveau rapport cyclique*/
unsigned int Force (double rap)
{
unsigned int result_force;
//prendre la valeur absolue de rap:
if (rap >= 0.0) ;//si oui il est negatif alors ne rien faire
else       //le rapport est negatif il faut donc le multiplier par -1 pour le mettre en positif
           {rap=(-1.0)*rap;
           }
/*Algorithme de régulation proporionnel , cet algorithme est définit dans le rapport*/
/*Il peut être modifié*/
/*PWx=40000-(rap*571-17100)*/
result_force=rap*571;
result_force=result_force-17100;
result_force=40000-result_force;
return result_force;
}
/* -------------------------FIN de FORCE           ----------------------------     */



/* ------ main Program:------------------------------------------------------------              */
```

```
void main (void)
{
        double Photo1;          //tension  de photodiode 1
        double Photo2;          //tension  de photodiode 2
        double ecart;           //ecart en pourcentage entre les 2 photodiodes

  DP71  = 1;            /*Le port P7.1 est mis en sortie                          */
                        /*si on avait mis DP7=0  ca aurait mis ce port en entrée  */
  DP72=1;               /*Le port P7.2 = sortie                                   */

  PM1   = 1;            /* PWM-Channel 1 Center Aligned Mode  sur P7.1            */
  PM2    = 1;           /*idem pour channel P7.2                                  */
  PP1   = 40000;        /* Load period register  : permet de définir la période de la PWM*/
  PP2   = 40000;
                        /* PW1 et 2 sont mis a 40000 au debut pour que le moteur soit arrêté */
  PW1   = 40000;        /* Load puls width register                                */
  PW2   = 40000;
  PEN1  = 1;            /* P7.1/POUT1 autorise le signal de sortie                 */
  PEN2  = 1;

/*precaution: les 2 PWM= arrêtées*/
  PTR1=0;
  PTR2=0;

  while (1)         /* tourne a l'infini*/
  {
        // Lecture des 2 tensions en entrée
        Photo1=ad_conv(0);              //P5.0
        Photo2=ad_conv(1);              //P5.1
                if((Photo1<400.0) && (Photo2<400.0))    //test si le signal d entrée est faible si oui: faire tourner le moteur
                  {                                     /*100 correspond à un signal de 0.5V                    */
                                                        /*remarque: ce test est dangereux car si le signal est faible le */
                                                        /* moteur ne trouvera jamais le signal                  */
                        //tourne sens aiguille montre
                        PW2=40000;      //freine et arrete la PWM2
                        PTR2=0;
                        PW1=30000;      //va lentement
                        PTR1=1;//lance la PWM1
                }
                else  // c est donc qu'un signal a été détecté
                {
                        /*Calcul de l'ecart en pourcentage*/
                        ecart=Photo1-Photo2;
                        ecart=ecart*100;
                        ecart=ecart/1023.0;                     // -100 <= ecart <= +100

                        if( (ecart <=30.0)  && (ecart >= -30.0) )
 /*si oui : stop les 2 PWM car le capteur optique est aligné en face de la source*/
                        {
                                PW1 =40000;     //Stop PWM1
                                PW2 =40000;     //Stop PWM2
                                PTR1=0;
                                PTR2=0;
                        }
                        else if( ecart > 30.0 )         //tourner sens  aiguille montre, et lancer PWM1
                        {
                                PW2 =40000;     //arrete PWM2
                                PTR2=0;

                                PW1 = Force(ecart);     /*calcul la valeur à recharger dans PWM1 */
                                                        /* Plus l'ecart est grand et plus la vitesse sera grande */
                                PTR1=1;//active PWM1
```

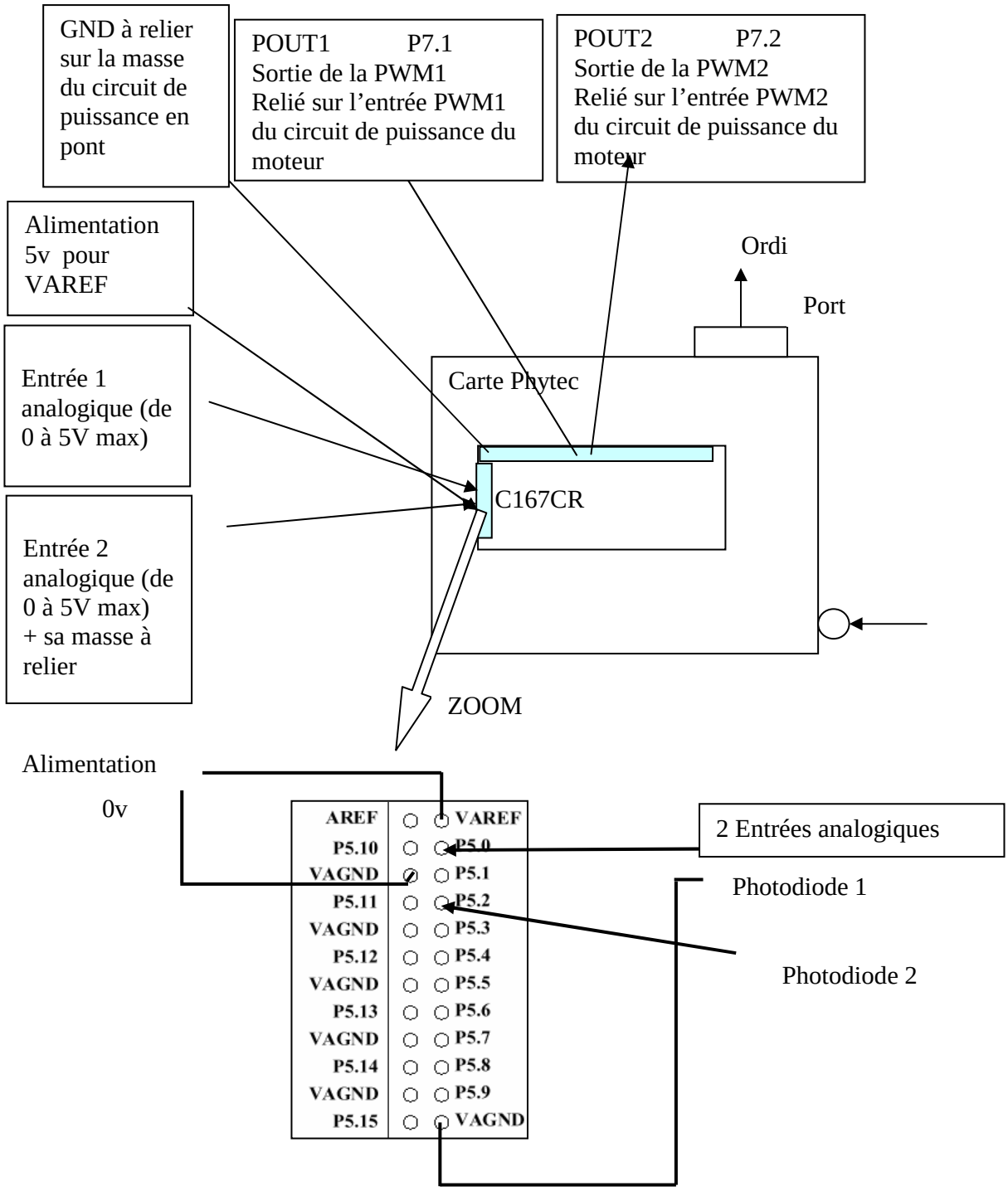
```
                        }
                        else if(ecart < -30.0 )         //tourner sens  inverse aiguille montre lancer PWM2
                        {
                                PW1=40000;
                                PTR1=0;
                                PW2= Force(ecart);
                                PTR2=1;
                        }
                }//fin else
  }//fin du while
}
/* -- End of program ------------------------------------------------ */
```

Ce programme marche correctement en pratique: il permet de bien centrer le capteur optique en face de la source infrarouge.

En vue d'une implémentation des 2 autres photodiodes voici les 4 unités PWM disponibles sur le microcontrôleur.

Table 7-7 Alternate Functions of Port 7

Port 7 Pin	Alternate Function	
P7.0	POUT0	PWM model channel 0 output
P7.1	POUT1	PWM model channel 1 output
P7.2	POUT2	PWM model channel 2 output
P7.3	POUT3	PWM model channel 3 output


GND à relier sur la masse du circuit de puissance en pont
POUT1 P7.1
Sortie de la PWM1
Relié sur l'entrée PWM1 du circuit de puissance du moteur
POUT2 P7.2
Sortie de la PWM2
Relié sur l'entrée PWM2 du circuit de puissance du moteur
Alimentation 5v pour VAREF
Ordi
Port
Entrée 1 analogique (de 0 à 5V max)
Carte Phytec
C167CR
Entrée 2 analogique (de 0 à 5V max) + sa masse à relier
ZOOM
Alimentation
0v
AREF
P5.10
VAGND
P5.11
VAGND
P5.12
VAGND
P5.13
VAGND
P5.14
VAGND
P5.15
VAREF
P5.0
P5.1
P5.2
P5.3
P5.4
P5.5
P5.6
P5.7
P5.8
P5.9
VAGND
2 Entrées analogiques
Photodiode 1
Photodiode 2

8.4. Problèmes rencontrés

Plusieurs problèmes se posent:
Si aucun signal n'est détecté le moteur continuera à tourner (à l'infini). Pour résoudre ce problème il serait utile de vérifier quand le moteur à réaliser 1 tour. Des capteurs hall sont disponibles dans l'axe mais le temps ne m'a pas permis de les utiliser.

Le programme tourne toujours en boucle, peut-être que l'utilisation de timer aurait été préferrable mais dans un sens cela ne nous gêne pas car la tâche du microcontroleur n'est dévouée qu'à ce programme. L'utilisation de timer (et donc d'interruption) serait une optimisation. La lecture sur les photodiodes s'effectueraient tous les x millisecondes.

L'implémentation des 2 autres photodiodes permettraient de centrer le capteur optique selon l'élévation.
Cela n'a pas été réalisé: et cela ne devait être fait qu'en cas où le temps me le permettrait. De plus il semble qu'une photodiode ne marche pas correctement. Le moteur qui permet de régler la position selon l'élévation a été réparé à 3 semaines de la fin du projet. C'est à ce moment que je réalisais le circuit imprimé sur PCAD.

9. Conclusion

Ce projet a été très instructif et intéressant et ce sur de nombreux points. Il a été l'occasion de mettre en pratique un montage électronique qui traite le signal provenant de photodiodes. La partie pré amplification permet de recevoir des signaux allant jusqu'à 80-100m. Mais en ajoutant la partie filtrage et redressement le signal est atténué et la distance de détection diminue.
La partie programmation sur le microcontrôleur s'est bien déroulée, les exemples fournis avec la carte de développement m'ont permit de rapidement prendre en main le logiciel. Je connais désormais la programmation du microcontrôleur C167CR en langage C ainsi que le logiciel Tasking.
Les principaux problèmes rencontrés viennent du montage électronique avec notamment l'atténuation du signal.
La partie régulation du moteur azimut est réalisée de manière simple, seul un régulateur proportionnel est implémenté.
Il a été intéressant d'utiliser les logiciels tel que PSPICE et PCAD. La partie PCAD n'est cependant pas tout à fait terminée, il faudrait juste vérifier s'il y a des erreurs.

Ce projet a été l'occasion de traiter différents domaines :
Comme l'électronique, l'optique, le traitement de signal, la commande de moteur à courant continu, la régulation (même si le régulateur utilisé est simple), le domaine du Microprocesseur…

Il reste donc à implémenter les 2 autres photodiodes pour réaliser le centrage selon l'élévation. La difficulté principale sera de réaliser le programme. La partie électronique peut encore être améliorée.

10. Annexes